THE ACQUISITION OF MORPHOPHONOLOGY UNDER A DERIVATIONAL THEORY: A BASIC FRAMEWORK AND SIMULATION RESULTS

by

Ewan Dunbar

Department of Linguistics
University of Toronto

Abstract

The Acquisition of Morphophonology Under a Derivational Theory: A Basic Framework and Simulation Results

Ewan Dunbar
Department of Linguistics
University of Toronto
2008

Since Rumelhart and McClelland 1986, the alternations found in the simple morphophonology of the English past tense have been run through numerous learning algorithms. These algorithms generally perform well at learning these very easy alternations, but fail to reflect a basic assumption about how grammar works held by many morphologists, namely, that affixation has a special status, different from other kinds of morphological changes. They also often do not address the question of how the learner restricts alternations like ***lie–lay*** *to a single sense of* ***lie*** *("lie down," and not "prevaricate") while still learning general patterns that apply to novel words regardless of their meaning. More generally, the morphophonological mappings deduced by previous morphological learning systems have never much resembled the kinds of grammars proposed by morphologists. In this paper I remedy this situation by constructing a system that deduces two kinds of rules—internal changes and suffixes—and applies them to words derivationally; it automatically finds rules that are not generally applicable and marks them as only applying to an arbitrary class of words. The model is largely compatible with developmental and psycholinguistic evidence; to the extent to which it is not compatible with this evidence, it raises interesting questions about how human morphophonological generalizations are best characterised.*

Acknowledgements

No one has tolerated my hours of writing and time spent in a windowless room away from home more patiently than my wife, Smiljka: certainly not me, because the patience game changes entirely when there is something waiting at the end, and even if she had had a Forum paper of her own waiting at the end, I know she would have found this to be small comfort.

My advisor, Elan Dresher, who was not infrequently given sporadic and incomprehensible drafts that I had already begun to rewrite, was patient enough to actually read these, and to give immeasurably helpful feedback (without which this paper would be a sporadic and incomprehensible draft, of course), and my second reader, Suzanne Stevenson, put an indispensable nonsense filter on the final draft. I also made the most of my time spent in the doctoral application process—which haunted this paper from before it was conceived—in pithy conversation with several other wise people (who know what they are talking about and who know who they are). All remaining factual errors, logical errors, misconstruals, misunderstandings, failures to be insightful, and all-around guff, is entirely my fault.

Finally, this paper would never have happened without the rest of the Forum class: Mike Barkey, Laura Baxter, Ailis Cournane, Vanessa Crandall, Ann Dolan, Yumiko Gondaira, Sofia Kanibolotskaya, Emma Lawson, Meagan Louie, Christina Marshall, Mark Pollard, Michelle St Amour, and Lyn Tieu. Indeed, it makes almost as much sense to me to add this as it does to add that I could not have written this paper without myself (everyone's dream, I suppose) because imagining this year without them is not really imagining this year, and this paper could not have been written without this year. My only hope is that, by now, at least some of you know me well enough to know how sincerely I mean it when I say a thing like that.

Contents

Chapter 1

Introduction

1.1 Grammoids: Learning Linguistic Lessons

Language acquisition is often conceived of as a *selective* learning task (Lightfoot 1982, Lightfoot 1991). Starting with Chomsky and Lasnik 1977, many authors began suggesting that most of syntactic acquisition takes place by choosing between a number of "settings" (usually two) for each of a finite collection of "parameters." Each parameter value leads to complex, biologically fixed syntactic behaviour.

This way of conceiving of learning was presumed to lead to more restrictive theories of the human genetic endowment for language than would casting learning as *constructive*. Earlier syntactic theories (popularized in Chomsky 1957, Chomsky 1965) suggested that grammars could contain arbitrary transformational rules like (1).

(1) $$NP_1 - Aux - V - NP_2 \quad \rightarrow \quad NP_2 - Aux + be + en - V - by + NP_1$$

(2) $$\begin{array}{ll} & [_{NP_1} \text{ The cat }] \text{ Prog}_{Aux} \text{ eat}_{V} [_{NP_2} \text{ the cheese }] \\ \rightarrow & [_{NP_2} \text{ The cheese }] \text{ Prog}_{Aux} \text{ be+en eat}_{V} \text{ by } [_{NP_1} \text{ the cat }] \end{array}$$

This transformation looks for strings of a particular syntactic composition (active transitive sentences) and rearranges their components (so that they are passive)—NP_1, previously the subject, is moved into a *by*-phrase, NP_2 becomes the new subject, and passive morphology is added, as in (2) (putting the verbal morphology in the right place takes place later, we assume).

There are clearly infinitely many rules with the general form that (1) has, namely, rules that reorder, delete, and add, arbitrary things to/from some arbitrary syntactic object. If arbitrary rules are possible, surely the learner must posit each expressly, not “select” from an existing list of *all* the possible rules. The idea that language acquisition could take place by selection was a major shift.

Although the selective theory is sometimes considered a major departure from previous views of Universal Grammar, it is perhaps more precisely thought of as a characterisation of the learning mechanism, the procedure that searches for the correct set of grammatical facts given the primary linguistic data. Searches carried out by enumerating all the cognitively allowed possibilities are selective—plausible, we assume, when the number of facts to be learned is relatively small, and surely only when the set of possible grammars is finite—hence the parametric view of UG. In larger hypothesis spaces, we assume that searches orient themselves by formulating some hypothesis “based on” the input in some way—like an initial guess at a transformational rule, based perhaps on some assumed semantic relation between two attested syntactic constructions—before exploring other hypotheses, presumably by means of small changes to the previous hypothesis. This is the kind of learning we call constructive.

It is not clear to me that we can find a precise demarcation line distinguishing selective from constructive acquisition procedures, but they are clearly different in some way. Nevertheless, the focus on selective learning, thought of as being fundamentally different from constructive learning, has led to certain gaps in our terminology in linguistics. In discussions of selective learning, the term *parameter* is used to refer to one member of the finite set of questions the learner needs to answer; each possible answer is a *parameter value*. Facts must also be acquired in a constructive learner, however, and yet, no term exists for the constructive equivalent of a parameter. In the type of syntactic theory discussed above, for example, the learner must find the correct set of transformations from among all the possible sets of transformations. The set of transformations in the grammar is thus a fact to be learned—but it is unnatural to call it a “parameter.” Similarly, if a learner had some idea of what the set of transformations was, but there was some sense in which it was simultaneously entertaining several “versions” of the same transformation, then the correct version of the rule would be another fact to be learned—but surely not a “parameter.”

In this paper, we will need a term that can refer both to facts to be learned constructively and to facts to be learned selectively. I will use the term *linguistic lesson*—parameters are linguistic lessons when our language acquisition device is selective. The possible answers for a linguistic lesson I will call *grammoids*—when we call our linguistic lessons parameters, we call these parameter values.

We need such terms because the topic of this paper is the acquisition of morphophonology—the phonological forms of morphemes and the alternations they induce in each other when combined—not a problem that is amenable to "selection" when we consider the number of possible morphemes that can be articulated by the human speech apparatus, and the number of morphophonological alternations attested in these morphemes in the world's languages. In the current study, we will build a learner for the English past tense. This learning task has been explored many, many times, from various perspectives (Rumelhart and McClelland 1986, Pinker and Prince 1988, MacWhinney and Leinbach 1991, Daugherty and Seidenberg 1992, Plunkett and Marchman 1993, Ling and Marinov 1993, Ling 1994, Plunkett and Juola 1999, Pinker 1999, Yang 2002, among others). The purpose of the current study is not to compete with these models or to fuel any of the numerous debates that have surrounded them.[1] Rather, the English past tense will be taken as a convenient data set to explore some of the issues that arise in constructive learning of morphophonology. In this introductory chapter, I will lay the foundations for this model.

1.2 Learning Phonology

Language acquisition must surely include a good deal of revision. Every linguistic lesson to be learned seems to require the learner to have learned a slew of other lessons first. Most obviously, if any lesson about a lexical entry has not been learned properly yet, the learner will probably make a bad guess about the rest of the grammar too. In phonology, this means that underlying representations must be reasonable if the rest of the grammar is expected to be right—and, regardless of how we choose to formalize phonological processes, if we endorse the traditional conception of a single, stored, underlying form (see Chapter 2 for some relevant discussion), that form must sometimes deviate from its surface phonology (otherwise there would be no phonology). But how is the learner to learn an underlying form without knowing anything about the phonology of its language? It must start with a guess which it is capable of revising.

In this paper, I will ignore the crucial issue of underlying form learning, but it is worth working out what would go into solving it. Clearly, we would need two things: first, a morpheme

[1] See Chapter 2.

segmentation algorithm—without morphology, there can be very little phonology, and thus little to say about underlying forms. Second, a criterion for when and how it should revise hypotheses about underlying forms. Both of these processes would be *constructive* in the sense of the previous section, to the extent that the selective/constructive distinction is meaningful (there are surely an infinite number of possibilities in the universe of underlying forms).

We can imagine what the second of these, a criterion for revising underlying forms, might look like: if we work backwards from the surface form using our current knowledge of the language's phonological processes, we might plausibly find a better underlying form than the one we have. I will not address the issue of underlying form learning in this paper, but one can easily imagine adding such a component to the model described here.

The first device the learner must employ for learning underlying forms is somewhat more complicated. In order to do any morpheme segmentation on an input form, the learner must surely have encountered at least some of its constituent morphemes before; knowledge about the meaning of the utterance would surely also help the learner to infer that more than one morpheme is present. While we could come up with some simple strategies for guessing at where forms are concatenated, the segmentation process must be somewhat more complicated than that. In particular, it must not always treat the constituent morphemes as simply concatenated strings. Rather, it must sometimes substitute an existing underlying form for one of the morphemes—for example, a learner whose target grammar had intervocalic voicing of *d,* and who recognized in the surface form *bada* a suffix *-a* and a stem meaning something suspiciously similar (presumably in meaning) to a known stem *bat,* would eventually need to substitute *bat* for surface *bad* in the decomposition. Otherwise, no progress would ever be made, and underlying forms would just be memorized surface forms. Since the problem involves meaning, it is somewhat more complicated than simply revising stored forms based on existing phonological processes. Nevertheless, we can assume that learners will deduce *some* underlying forms each time they hear an utterance. We can tentatively explore the problem of learning the stored–surface mapping by assuming that learners are able to deduce the correct underlying forms immediately. This is reasonable in the case of the English past tense, where (what is probably) the stem—the unmarked present tense—is frequently attested in isolation. The issue of how learners work out which form is the stem remains open. In the case of concatenative morphology, this problem is often addressed in research on morpheme segmentation (for example, Goldsmith 2001); for an interesting new approach that seems to work in the general case, see Chan 2008.

The focus of this paper is to explore some of the problems that arise even when we narrow our attention just to finding the mapping between known underlying forms and their associated allomorphs. In that (simplified) case, we can imagine that part of the learning procedure takes as its input a pair: an underlying form, and the attested surface form. We should distinguish between the *input* to some part of the learning process—say, one dedicated to learning the underlying–surface mapping—and the utterance. What exactly the input to each part of the learning procedure is will depend greatly on what we think the learning procedure looks like.

Suppose that the learner (the term we will use from now on to refer to this restricted part of the learning procedure) gets as input the underlying–surface pair $\langle \text{ila}, \text{ela} \rangle$. Assuming for now that the learner believes it is dealing with words (# boundaries), and that it is learning an underlying–surface mapping in the form of rules, it might then posit the simple rule in (3).

(3) $\text{i} \rightarrow \text{e}/\#\_\text{la}\#$

The rule shown in (3) is correct, but if we filled our grammar with rules like this, we would begin to wonder what rules were for. The problem with this rule is that it is not very general. In fact, it is almost as specific a rule as we could imagine. Suppose we wanted to find a more general rule. On the basis of only one example, any generalization we might make could be dangerous. Even the only slightly more general rule in (4), for example, might be wrong—for example, if we later got $\langle \text{bilap}, \text{bilap} \rangle$ as input.

(4) $\text{i} \rightarrow \text{e}/\_\text{la}$

One way of dealing with the issue of overgeneral rules might be to later remove or otherwise disprefer rules if they are found to be too general, and substitute more specific ones. A simpler way of dealing with the problem of overgeneral rules, however, is to consider less general rules first. Many authors have therefore suggested conservative phonological learning, as shown in (5), which we will call *Minimal Generalization Learning* (MGL).[2]

[2] It was suggested in Pinker and Prince 1988 and implemented in Yip and Sussman 1997 (there as a batch algorithm) and Molnar 2001; the name *Minimal Generalization Learning* was first used in Albright and Hayes 1999, referring to an algorithm which includes an extra, batch ranking step which is irrelevant here; Albright 2002, Albright and Hayes 2002, Albright and Hayes 2006, and Albright in prep., among other papers by the same authors, make use of the same system. The finite-state-transducer-induction algorithm Gildea and Jurafsky (1995) apply to phonology does not generalize beyond its input, although there the implementation details make the insights more relevant to the study of transducer induction than of phonological learning. The phonology-learning system of Ling and Marinov 1993, Ling 1994 also has a rather different character.

(5) On input $\langle s,t\rangle$:

a. If there is a change $\alpha \rightarrow \beta$ in $\langle s,t\rangle$ so that $s = q\alpha r$ and $t = q\beta r$, posit $\alpha \rightarrow \beta/\#q\_r\#$.

b. If we have already posited a rule $\alpha \rightarrow \beta/Q\_R$, generate a new rule $\alpha \rightarrow \beta/Q'\_R'$, so that Q' (R') is the most restrictive statement we can make of the set of environments including both $\#q$ and Q ($r\#$ and R).

This learner, after hearing ⟨ila, ela⟩, would posit the rule in (3), repeated here as (6). After hearing ⟨fimp, femp⟩, the learner would then extract the rule in (7) and combine it with (3) to a rule like (8), assuming that the learner states sets of environments by aligning the phonological strings and combining them using some straightforward feature system (crucially allowing no intersection between # and #f or between a# and p#).

(6) i → e/#_la#

(7) i → e/#f_mp#

(8) i → e/_[+cons, +son]

The description of this learning paradigm as *conservative* and *minimally-generalizing* comes from the fact that the learner does not extend the alternation to any more environments than it needs to to account for the data. It would not extend the rule to non-sonorant consonants, for example, without some evidence of the alternation before a non-sonorant consonant. This kind of learning will avoid overgeneralization to a large degree.

One obvious case in which overgeneralization will nevertheless happen is when rules have exceptions. Another is when rules are ordered. This raises the possibility that the [m] in [femp] is derived from underlying /n/. If the learner does not know enough about the rest of the system, and has not learned the correct underlying form yet, it might take ⟨fimp, femp⟩ as input and derive the rule in (8), possibly wrongly (for example, if the i → e alternation really only applies before /l/ and /n/, thus allowing the [im] sequence barred by (8)).

The simple morphophonology of English being the testing ground for most phonological learning systems, previous implementations of algorithms like this have generally only had to deal with the specific problem raised by exceptions or by the presence of additional idiosyncratic processes (like the one that derives *ring–rang*, which should block the application of the regular *-d* rule), not by rule interaction more generally. This allows creative solutions to the problem of overgeneralization which would be impossible otherwise. For example, the learner in Molnar 2001 simply associates a list of forms with each rule—even the general *-d* rule—not a workable solution if our goal is to simulate real-life learning. The learner first presented in Albright and Hayes 1999, on the other hand, does not deal with these problems at all—the authors assume that, although speakers have rules for irregular past tenses like *ring–rang*, they do not use them for production, and memorize these forms in addition (see Chapter 2).[3]

In this paper, I will assume that Minimal Generalization Learning is correct, regardless of how realistic that assumption is. Holding this part of the learning mechanism constant, the problem will be that of adding the right mechanisms to deal with the relevant rule interactions and with exceptions. The main focus will be on adding the kind of features needed to ensure that the English strong verbs—*ring–rang*, *bring–brought*, *cut–cut*, and all other English verbs that do not (only) take *-d* in the past tense—will eventually be correctly inflected. This goal presupposes a particular view of the architecture of morphophonology. The assumptions I make will be laid out in section 1.4 and discussed at greater length in Chapter 2. Before discussing these assumptions, however, I will attempt to complete the picture of the assumptions I make about learning: MGL is a way of doing constructive, not selective, learning, in the intuitive terms laid out in the previous section. We will also need some tools for doing selective learning; it is to those that we now turn.

1.3 Stochastic Grammoids

Children's speech is variable. During a single session, two-year-old Abe (Kuczaj 1977) referred to his horn once as *mine horn*, and then as *my horn* just a few minutes later. Two-year-old Eve (Brown 1973) told her mother what she wanted thirteen times in one session—eleven of those times, she said *I want . . .* (*I want some more tapioca*)—but, for one reason or another,

[3] While this learner does not address the problem of interacting rules, however, it is constructed to assign (variable) inflection to nonce stems. When presented with *spling*, for example, it is supposed to mimic English speakers' behaviour, favouring both the regular *-d* alternation and the ɪ → ʌ alternation (as in *sting–stung*) over, say, a rule adding [ɔt] (as in *bring–brought*). It does this keeping all the rules it has ever posited in the grammar, and treating the situation something like a selection problem: the learner collects some simple statistics about how well its rules work, and ranks them, to determine *which* of the various versions of (say) the [ɔt] rule it prefers—a very general one applying to all words (which would presumably be deduced because the words that take this alternation, *bring*, *fight*, *catch*, *seek*, *think* and *teach*, seem to have very little in common) or a collection of very specific ones (one for each of #brɪŋ#, #fajt#, #kætʃ#, #sik#, #θɪŋk#, and #titʃ#). The learner does, therefore, deal with overgeneralization to a certain extent, but is not sufficient if we intend to describe speakers' knowledge about existing verbs—for example, that the past tense of *catch* is definitely *caught*, and not **catched*, and that the reverse (just as definitely) holds of *latch* (past tense *latched*, not **laught*).

she twice said *Want*... (*Want tapioca*). These grammatical mistakes are temporary ones—the children use the correct forms both before and after. How can this be?

Another way of saying that children are making grammatical mistakes—as long as they are making true errors in *grammar*, rather than mistakes which can be well explained by general processing or production factors—is to say that they are using incorrect grammoids. Our goal in the study of language acquisition is to find the algorithm by which children deduce the correct grammoids for their language, and to do it in a way that explains these variable errors. If children are making occasional, random slips whereby they use a wrong grammoid, there are two ways of building a learning model to suit, only one of them reasonable.

The unreasonable approach would be to attempt to *explain* each of these minor changes in the child's grammar in terms of the input received by the child. We would develop a learning algorithm which would occasionally—but predictably—make very small changes to its state without having heard any input; if a child were observed alongside its doppelgänger—an identical child, getting the exact same string of inputs, attempting to say the exact same things at the same time—we would expect the child and the doppelgänger to make the exact same mistakes, at the same time. This would be a *deterministic* approach, and it would be a daunting task.

The simpler approach, however, would concede that there are certain kinds of variability in children's speech that cannot be explained any more than we would attempt to explain each microdeviation in the path of a billiard ball. Rather, we would treat the child's output as unpredictable, or *stochastic*. Children's productions follow some probability distribution, so that, where there might be only one way for an adult to say something, for the child certain grammoids are more likely than others. When the child says that thing, it outputs one of the possible forms, but there is no guarantee which—only probability. The child and its doppelgänger would make the same *kinds* of mistakes, at similar rates, but we would never try to develop a theory of why the child made a particular mistake at a particular time that went beyond quantifying the child's propensity to make that sort of error at that time. A theory of learning would be an explanation of why the child's propensity to make various errors drops off.

One way to implement this, suggested by Yang (2002), is to assume that the learner can be uncertain about what the correct grammoid solution to a linguistic lesson is, and that children's unpredictable outputs are largely due to this uncertainty. In other words, we assume that some linguistic lesson may be simultaneously believed by the learner to be solved by two or more different grammoids, each with some belief strength. The learning algorithm alters belief strengths, and belief strengths are directly reflected in children's productions. I will call this assumption the *Stochastic Grammoid Theory (SGT).*

For example, if an adult says *It is raining* rather than *Raining*, we assume it is because that adult's belief in the grammoid $+$*pro-drop* has zero or negligible strength (ignoring variation); a child, on the other hand, might say *It is raining* only sixty percent of the time that it intends to say such a sentence, and *Raining* the remainder of the time—representing belief strengths by quantities on the interval $[0, 1]$, we conclude that the strength of its belief in the $-$*pro-drop* grammoid is 0.6 (and thus 0.4 for the $+$*pro-drop* grammoid*).*

An SGT-based theory of language acquisition has the advantage of making predictions about the quantity of certain utterances we expect to see in children's production. Yang's (2002) *Variational Learning* (VL) is one such theory. It holds that children learn grammatical lessons by a kind of weeding-out process, analogous to biological evolution, so that grammoids that are not sufficiently "fit" given the primary linguistic data become gradually dispreferred. Unlike the language learning systems usually called "genetic models," however, (for example, Clark 1992), Variational Learning uses a simple criterion for the fitness of a grammoid: does it allow us to derive the current input? Yang assumes that, after the learner hears an utterance and after it has attempted to determine what the speaker's underlying semantic, morphosyntactic, and phonological representations were (that is, what was meant), the learner then attempts to determine what utterance *it* might produce given the same intentions, given its current beliefs. It chooses all the necessary grammoids according to the probability distribution implied by its belief strengths for each linguistic lesson—just as SGT implies happens in production—then adjusts the strengths of its beliefs in the chosen grammoids depending on how fit they are deemed to be—that is, whether they will derive the attested utterance.[4]

[4] SGT holds that the overall rate at which children produce errors of a certain kind is evidence for "belief" in some grammoid not used in the ambient language that would lead the child to such an error, with a "strength" proportional to the rate at which such errors are produced. By far the simplest theory of why this would be the case holds that producing a sentence involves choosing some set of grammoids. If these choices are made according to the probability distribution induced by the strengths of the learner's beliefs in the various possible grammoids, then learners will make errors/correct productions at rates proportional to their belief strengths by sheer force of arithmetic—if having a belief strength of 0.6 for $+$*pro-drop* means a speaker will choose a $+$*pro-drop* grammar for production with probability 0.6, clearly we should expect 60% of that speaker's sentences to be made using a $+$*pro-drop* grammar. This is what SGT requires us to believe for production—but what about comprehension?

VL views learning as a process of covert production by the hearer of the same sentence the speaker intended to produce. In order to do this the hearer must know what the speaker's intention was—that is, it must have comprehended the sentence to some degree. The simplest inferences about learners' belief strengths based on their error rates would assume that their comprehension is perfect when they do learning—and that would lead us to the conclusion that learners have perfect comprehension before they have learned anything—and that makes no sense: the fact that I do not speak Ojibwe means I can neither produce *nor* comprehend grammatical Ojibwe sentences. It makes sense that belief strengths should affect comprehension in some way, but the simplest hypothesis, that comprehension depends on belief strength in exactly the same way that the theory says production does, cannot

With a sufficiently explicit theory of how the learner uses fitness information to adjust belief strengths, we can test our theory of learning and of UG (what grammoids are available to the learner to solve what linguistic lessons) by making predictions about how often children should mistakenly use certain incorrect grammoids, either by simulation or—in simple cases—analytically.

So far, Yang's studies of children's errors under the belief update scheme in (9) (the *Linear Reward–Penalty*, or L_{RP}, scheme) seem promising.

(9)

$$\text{Selected grammoid } G_i \text{ derives input:} \quad \begin{cases} B_{t+1}(G_i) = B_t(G_i) + \gamma \cdot (1 - B_t(G_i)) \\ B_{t+1}(G_j) = B_t(G_j) - \gamma \cdot B_t(G_j) \\ \text{(for alternatives, } G_j\text{, to } G_i) \end{cases}$$

$$\text{Otherwise:} \quad \begin{cases} B_{t+1}(G_i) = B_t(G_i) - \gamma \cdot B_t(G_i) \\ B_{t+1}(G_j) = B_t(G_j) + \gamma \cdot (\frac{1}{N-1} - B_t(G_j)) \\ \text{(for alternatives, } G_j\text{, to } G_i) \\ N = \text{total number of grammoids} \end{cases}$$

The L_{RP} update scheme *rewards* chosen grammoids when they are fit, and *penalizes* them otherwise. It *rewards* a grammoid G_i with belief strength $B(G_i)$ by increasing that grammoid's belief strength by $\gamma \cdot (1 - B(G_i))$ (a quantity linear in $B(G_i)$, whence the name; γ is the *learning rate*, a constant on $[0, 1]$, a parameter of the learning model); it must also then decrease the strength of its belief in all of the other candidate grammoids so that grammoid choices can continue to be made according to a valid probability distribution (probabilities must always add up to one; for a crash course in probability theory, see Appendix A). It *penalizes* a grammoid G_i with belief strength $B(G_i)$ by decreasing that grammoid's belief strength by $\gamma \cdot B(G_i)$ and increasing the strength of its belief in all of the other candidate grammoids to compensate. To see how the system works, consider the simple case wherein the learner must learn a single

be true. It is not unreasonable to think that there is more to comprehension: consider that the learner may hear many forms it cannot derive, but still have plenty of extralinguistic contextual information so that the meaning, syntax, and so on, is entirely transparent.

All this means that it is more complicated to predict what learners *understand* than it is to predict what they *do*. The simplest prediction we can make is that learners should reach receptive competence before they show reliable productive competence—and that is a generalization which is widely attested in child development (for example, Winitz *et al.* 1981). The simplifying assumption we will take up here is that we are consistently modelling a point in time *after* learners attain enough competence to deduce speakers' intentions correctly. See section 1.2 for discussion of this simplifying assumption in the context of morphophonology.

linguistic lesson which two grammoids might possibly solve—say, the widely hypothesized binary $\pm$*pro*-drop parameter, which regulates the production of sentences without overt subjects. Suppose it starts off believing in both $+$*pro*-drop and $-$*pro*-drop grammoids equally strongly (that is, each has a belief strength of 0.5). On each input,[5] the learner chooses a grammoid, either $+$*pro*-drop or $-$*pro*-drop, and simulates production in order to decide whether that grammoid would derive the attested input.

Suppose that the correct (ambient) grammoid is $+$*pro*-drop. This grammoid is taken to make it possible—though not *necessary*—for speakers to omit the grammatical subject of a sentence, except for diary drop (*Dear Diary—Went to the store today*, which can only have a first-person meaning) and imperative sentences, which we take to be licensed for both $+$*pro*-drop and $-$*pro*-drop. Whenever the learner chooses $+$*pro*-drop for its predictions, it will always be able to derive any sentence it might have heard. It will thus always reward $+$*pro*-drop (which, of course, becomes a problem if the ambient grammar is $-$*pro*-drop—see below—but is a positive thing here). If the learner chooses $-$*pro*-drop after having heard a sentence not permitted by that grammoid—like *Went to the store today,* in the case where the speaker intends to express that *someone else* (third person) went to the store—it will always penalize $-$*pro*-drop. If it chooses $-$*pro*-drop after having heard any other kind of sentence, it will reward $-$*pro*-drop, incorrectly, but, since such a sentence must also be licit in a $+$*pro*-drop grammar (since it was in the input), we expect that the learner will have better evidence for the correct grammoid than for the incorrect one. A sample course of acquisition is shown in (10) (where $\gamma = 0.1$).

[5] Or rather, on each input that the learner correctly understands—which we will pretend is the same thing (see fn 4 and section 1.2 above).

(10)

Our beliefs are…					
$B(+)$	$B(-)$	Then we hear…	And we choose…	Which is…	That means we…
0.500	0.500	*Went to the store* (meaning, *They went to the store*)	+*pro*-drop	Fit	Reward +*pro*-drop
0.550	0.450	*Climbed a tree* (meaning, *He climbed a tree*)	−*pro*-drop	Unfit	Penalize −*pro*-drop
0.595	0.405	*John climbed a tree*	−*pro*-drop	Fit	Reward −*pro*-drop
0.535	0.465	*Went to the store* (meaning, *I went to the store*)	−*pro*-drop	Fit	Reward −*pro*-drop
0.482	0.518	*Went to the store* (meaning, *They went to the store*)	−*pro*-drop	Unfit	Penalize −*pro*-drop
0.534	0.466	*Climbed a tree* (meaning, *He climbed a tree*)	−*pro*-drop	Unfit	Penalize −*pro*-drop
0.580	0.420	*Is raining* (meaning, *It is raining*)	+*pro*-drop	Fit	Reward +*pro*-drop
0.622	0.378	*Is snowing* (meaning, *It is snowing*)	+*pro*-drop	Unfit	Penalize −*pro*-drop
0.660	0.340	*John climbed a tree*	+*pro*-drop	Fit	Reward +*pro*-drop
0.694	0.306	*Went to the store* (meaning, *I went to the store*)	+*pro*-drop	Fit	Reward +*pro*-drop
		⋮			

(10) shows us a few things. First, the learner receives two inputs consistent only with +*pro*-drop.

As a result, it begins to prefer that grammoid regardless of which grammoid is chosen. The next two inputs are compatible with both possible grammoids (the second is English-style diary drop). If it chooses the incorrect grammoid, $-$*pro*-drop, it will reward it, incorrectly—but after getting enough unambiguous evidence in favour of $+$*pro*-drop—like the following four inputs—the learner will be more likely to choose $+$*pro*-drop after hearing ambiguous sentences (the last two sentences). If it chooses the $+$*pro*-drop grammoid after hearing such a sentence, it will reward it, correctly. The problem of ambiguous evidence seems to be straightforwardly solved for a Variational Learner (using a "guessing" strategy in the sense of Fodor 1998). In fact, as Yang points out, we can prove that the learner will not be swayed by ambiguous evidence if we assume the L_{RP} system. If we view the PLD as a random stream of inputs with a fixed distribution over time—so that the probability of encountering a particular kind of evidence at a particular time is the same regardless of when that point in time is—then we can state, in terms of input frequency, a quantity that, on an average input stream, the learner's belief in some grammoid G_i will get arbitrarily close to as time goes by. This is shown in (11) (see Narendra and Thatachar 1989 for the proof, put in more general, learning-theoretic terms).[6]

(11)
$$\lim_{t\to\infty} E[B_t(G_i)] \quad = \quad \frac{c_i^{-1}}{\sum_{1\leq j\leq N} c_j^{-1}},$$

where c_k is the probability of encountering an input that G_k will not derive, N is the total number of possible grammoids for that lesson, including G_i.

The formula in (11) is stated in terms of input frequency—in particular, the probability of encountering an input unfit for each grammoid G_i—call this G_i's *penalty probability*, c_i, in that environment. The case of interest, where there is some grammoid which is correct for all inputs, is exactly the case where one grammoid—say G_1—has penalty probability $c_1 = 0$. We would like it to be the case that, on average, belief in this grammoid will over time get arbitrarily close to one. We cannot compute $\lim_{t\to\infty} B_t(G_1)$ directly from the formula in (11) in this case, however, because, when $c_1 = 0$, (11) works out to $c_1^{-1} = 0^{-1}$, which is undefined. We must instead ask whether, as c_1 gets closer to zero, the quantity in (11) gets arbitrarily close

[6] E denotes the expected value (average) function from probability theory. The lim, or limit, function, familiar from calculus, gives the value that a function gets arbitarily close to as its parameter approaches some value, or, in this case, gets arbitrarily large. For a crash course in probability theory, see Appendix A.

to one—that is, (12), holding c_i constant for $i \neq 1$ (and assuming $c_i \neq 0$ for $i \neq 1$—see below). This can be shown to be true using the basic properties of the lim operator (the simple proof is omitted here).

(12)
$$\lim_{c_1 \to 0} \frac{c_1^{-1}}{\sum_{1 \leq j \leq N} c_j^{-1}} \quad = \quad 1$$

The VL theory is thus a workable, simple way of dealing with the problem of ambiguous evidence—at least under the L_{RP} scheme, we can prove that a VL learner will eventually select the grammoid with unambiguous evidence, without the learner having to know in advance whether some piece of evidence is ambiguous.

Note that VL can help us with the problem of ambiguous evidence, but it cannot solve every problem. The only way in which the learner's beliefs can be adjusted is in response to the grammaticality predictions of some particular grammoid on some input. If there are *two* grammoids compatible with the input for some linguistic lesson, there will be no inputs forcing the learner to use one or the other. The $-$*pro*-drop grammoid is a famous apparent instance of this configuration. All sentences generable by this grammoid are compatible with the alternative, $+$*pro*-drop, because all $+$*pro*-drop means is that one can *optionally* drop the subject. It is not entirely clear what would be predicted in this case, but, at any rate, there is no reason to think the acquired grammoid would be the correct one (simulation confirms this).

A further complication comes from the issue of noise. If a small but consistent stream of counterexamples to the correct grammoid is present in the input (as it likely will be in real life), it is easy to derive a more general result than (12) showing that, under the assumption of L_{RP}, this will result in some residual strength for the incorrect grammar, proportional to how much of the evidence for one grammoid or the other was for that one. This is a serious issue in these situations, because there will be no evidence at all to counter the noise.

There are two ways to get around this possible limitation: first, we may suppose that there are no such situations (for example, if the $+$*pro*-drop grammoid rules out expletives, as in *It is raining*, which are licit in a $-$*pro*-drop, as seems to be the case). Second, we may suppose that some separate mechanism, which can be studied independently of the VL system, is responsible for working out these cases. I will assume for concreteness that something like this is true.

Assuming VL, we find it has some further advantages. Since it assumes SGT, for example, it has the distinct advantage of allowing quantitative predictions about children's errors straightforwardly. For example, it is possible to make a prediction about the relation between the rate at which a child will use the correct grammoid for a binary-valued parameter, and the penalty probability (see above) of the alternative. An approximate prediction of this kind is shown in (13), an approximation to G_1's belief strength at time t in terms of the penalty probability, c_2, of the alternate grammoid, G_2, and some arbitrary constants K and L ($E[\cdot]$ stands for the expected value operator, which gives an average over all possible cases, the best approximation we can get if we do not know which cases are most likely; see Appendix A for further explanation of the notion of expected value and for proof of this result). Even if we cannot work out exactly what this ought to mean for the learner's beliefs, This allows us to make rough predictions about the number of errors we should see coming from children over some time period based solely on input frequencies.

(13)
$$E[B_t(G_1)] \quad = \quad 1 - K(1 - L \cdot c_2)^t$$

A very simple prediction we can make given (13) is in a situation in which the non-target grammoid G_2 is known to have very little evidence ruling it out in some language—that is, a situation in which the penalty probability of G_2, c_2, is very small, but non-zero. We would predict late acquisition of the target parameter value G_1 compared to other parameter values with larger penalty probabilities (drawing a graph of this function, or substituting a few values of c_2, for some K and L on $[0,1]$, with K not too close to 1 and L fairly close to zero, and some not-too-large t, should convince the reader of this).

Just such a prediction is taken up in Yang 2002. The author supposes that there is a binary verb-placement parameter (call it ±V2, for *verb second*) such that, given that a number of other parameters are set correctly, one value of this parameter, $+$V2, will force the verb to come second in a sentence, allowing Subject–Verb–Object (SVO) order, Object–Verb–Subject (OVS) order, and XP–Verb–Subject–Object (XVSO) order, where XP is some adjunct, in transitive sentences. The $-$V2 setting, on the other hand, will, given that the same other parameters are set in the same way, allow Subject–Verb–Object, XP–Verb–Subject–Object, Verb–Subject–Object (VSO), and Verb–Object–Subject (VOS)—but *not* Object–Verb–Subject. It is a hypothesis (and an interesting one) that these represent two mutually exclusive possibilities, but *something* like this is just a fact, since, according to Yang, Dutch allows exactly the $+$V2 verb placements and Arabic exactly the $-$V2 ones.

(14)

	SVO	**XVSO**	**OVS**	**VSO/VOS**
+V2 (Dutch)	Yes	Yes	Yes	No
-V2 (Arabic)	Yes	Yes	No	Yes

As it turns out, in Dutch, 65% of matrix sentences are SVO, 34% XVSO, and only 1% OVS. By hypothesis, the learner has access only to positive evidence, so that the *absence* of VSO/VOS sentences does not count as evidence against −V2—but the *presence* of OVS sentences does. If we assume that the learner attends mainly to matrix sentences (the degree-0 theory of Lightfoot 1989), then the penalty probability of −V2 is 1%. This is far less evidence than, for example, learners of French have for the main verb placement in that language (before negation or adverbs—a so-called *V–T raising* grammar): according to Yang, about 7% of French sentences contain the sequence Finite Verb–Negation/Adverb, which is impossible without V–T raising. French V–T raising seems to be acquired by 1;8, so we expect Dutch children *not* to have acquired +V2 by this time, because they have seen fewer examples than French children have of V–T raising. Yang presents the results of a corpus study of a Dutch learner showing that this is, indeed, true: the proportion of the unambiguously −V2 sentences—that is, VSO/VOS sentences—in the child's output steadily declined, but, even by the end of the corpus at 3;0, 14–20% of the child's sentences represented the −V2 grammoid. This is a good result not only for SGT, but also for the study of UG. If the theory is correct, a language allowing SVO, XVSO, OVS, *and* VSO/VOS should be impossible. Similar results for the loss of verb-second placement in historical English and French can also be found in Yang 2002.

That this kind of result is possible is exactly why we want SGT. The promise of a theory of language acquisition is that it might bring more evidence to bear on more general questions about the language faculty. This is only possible if the theory makes some predictions about things that a theory of possible grammars cannot—like the course of acquisition. The more precise these predictions are, the better.

On the other hand, the configuration that allows us to make this particular prediction is somewhat precarious. First, we must assume that the learner has a two-way linguistic lesson to learn—and there are surely a variety of ways of fleshing out ±V2, whatever our theory of syntax, that are compatible with this claim.[7] Acquiring something like the meaning of *dog*, on the

[7] This applies to the study by Thornton and Tesan (2007), too—an interesting argument against VL. The authors claim that their data, which shows young learners moving very rapidly from near-total belief in one grammoid to near-total belief in another, is not compatible with VL. They present their reasoning quite explicitly: "There are two main predictions of the Variational model, as we understand it. One is that the two values of a parameter

other hand, is probably not a two-way choice. Even supposing that we thought VL was relevant here, (it only makes sense for selection problems, but we could easily imagine the learner maintaining more than one alternate hypothesis for a lexical-semantic fact, as with probably a variety of other sorts of linguistic lessons), the *n*-possibility grammoid generalization of the prediction in (13) is somewhat more complicated to work out than the binary-parameter version. More importantly, Yang's study presupposes that the learner has already learned its other linguistic lessons. That might be a safe assumption for certain cases, but in general, we cannot directly measure belief in a grammoid *P* by examining error rates; rather, we can only measure how often the learner uses the *full grammar* consisting of grammoids *P and Q and R*, and so on, in production. A full grammar is a conjunction of grammoids, which should be used by the learner with frequency proportional to the *product* of the beliefs in the individual grammoids, and this is an even more complicated quantity to make predictions about, as we will see in Chapter 2 (section 2.2.1). In fact, we cannot even be certain how to assign rewards and penalties in this, more realistic case: if the learner's chosen collection of grammoids does/does not derive the input, how does the learner know *which* one should be rewarded/penalized? (For some suggestions, see Yang 2002 and Chapter 3 below.) This complicates making any predictions even further. Ultimately, many of the predictions of any learning model will almost certainly need to be made by computer simulation, at the expense of the fullest possible understanding of the model's behaviour.

In this paper, I will apply SGT to the acquisition of English past tense inflection, outlining some of the most obvious issues arising when we attempt to construct such a theory and proposing

should be in conflict [*that is, they should both be attested with comparable frequency*–ED] early in the course of language development, at least for the majority of children. Second, to the extent that the incorrect value is favored over the correct value, the trajectory of development should be gradual." Since their data show a rather different pattern, they claim that VL cannot be correct, at least not for the linguistic lesson in question (whether negation appears in the head or spec of NegP)—but they are wrong to claim that VL *requires* competing grammoids to initially coexist with comparable belief strength. A *default value* for some linguistic lesson is surely possible if learners begin with only one possible grammoid available (we might then view part of learning that lesson as *constructive* rather than *selective*), or if their beliefs tend toward some grammoid initially. If this is the case, only one grammoid might be attested early on, leading us to see no "conflict." Of course, if this is the case, it is a fact that deserves some real explanation—and we should prefer one that makes predictions about other parts of acquisition rather than a stipulated, genetically-programmed order of parameter setting which does not permit us to relate acquisition to input in a meaningful way—which is, I think, a reasonable assessment of the explanation Thornton and Tesan propose. In any case, this is orthogonal to whether VL is correct or not. Their arguments against VL from the *speed* of acquisition are more convincing—though they are not right to suggest that the children they studied did not have gradual acquisition. They did, but it was very fast—and too fast, as they point out, to be acquired by the same means as Yang's baseline parameter, since, if the authors are correct, there is very little evidence against the incorrect grammoid—but the sudden onset of acquisition shows that there are clearly other factors in play in this particular case, factors which must be addressed before we can rule anything out.

some tentative solutions. In the final section of this chapter, I will discuss the assumptions about grammar that will guide this study.

1.4 Assumptions About Morphology

Issues of learning can only be discussed, of course, with some theory of the language faculty, and thus of what is to be acquired, in mind. In this study, I will take up a set of assumptions along the lines of Halle and Marantz 1993 and Noyer 1997 (although the most theoretically interesting assumption in those works, the late insertion of phonology, does not figure in here). Relevant to the current work are six assumptions, which I will list here and describe in more detail below.

First, I assume that, if a form appears to contain separate phonological pieces, it is generally because those pieces are stored separately and concatenated online (*full decomposition*). As a corollary, I also assume that concatenative morphology is, in general, processed by a mechanism distinct from internal-change morphology. Second, I assume that, if there are two different paradigms for the same grammatical function, and one appears to contain more phonological pieces than the other, then the missing pieces in the other paradigm are phonologically null but otherwise have the same status as their overt counterparts (*uniform decomposition*). This assumption has force only in conjunction with other assumptions about how grammar works that would make the notion "same status" meaningful; its place in the current study will become clear. Third, I assume that, if either internal or concatenative changes require a special phonological environment for licensing, those environments are well described by statements like those used in Chomsky and Halle 1968 (*SPE-style environments*). Fourth, both the concatenative and the internal-change mechanisms are derivational, consisting of a number of interacting processes (*ordered rules*). Fifth, internal changes and concatenation are also ordered with respect to each other, with internal changes that mark morphological information preceding concatenative changes (*ordered mechanisms*). Finally, processes which are licensed for a set of contexts that cannot be described by the ordered application of rules with simple, phonological environments are instead licensed (or barred) by a special marking associated with one of the concatenated pieces (*stem-marking*). I will now explain each of these assumptions more fully.

Full decomposition can be contrasted with *output-memorization*. Full decomposition states that forms that appear to be composed of different parts are indeed produced by concatening those parts, memorized separately, online. Output-memorization denies this, and claims that what appear to be the outputs of concatenation are really just stored in their concatenated form. Some obvious consequences fall out. We know that speakers are aware of patterns in morphology, and can follow these patterns to generate new forms—relevant here, for example, is the fact that English speakers will freely add what we will call the *-d* past tense morpheme (including its three allomorphs, [d], [t], and [ɪd]) to novel verbs. This makes sense on a full decomposition view: if making a past tense means concatenating a verbal piece and a past tense piece, then speakers ought to be able to swap out the verb for a new one. Believing in output-memorization means claiming another explanation: since speakers store (apparently) composed forms as complete units, separate from their uncomposed or differently-composed counterparts, the patterns that speakers are aware of are not used to generate forms online—in general they could be, but that is not what they are for. Thus, although speakers could not have stored any phonological form for the full past tense of *fingle*, they would if they could (but haven't heard it, so they can't). They could in principle hear it, furthermore, since the speakers around them are aware of the patterns in their stored forms and can produce new, similar forms. Both theories are possibly correct (although, not knowing exactly how "storage" in the brain works, we can imagine that they are more similar than we might expect). I will assume full decomposition.

Uniform decomposition means that null morphology is normal morphology. As noted, it is difficult to work out what the alternative to this would be without a notion of what counts as "normal morphology." In the late-insertion view associated with Halle and Marantz 1993—in which the phonological pieces that make up words are thought of as phonological "translations" (*exponents*) of morphosyntactic feature complexes, composed into words during the same procedure that composes words into sentences—this would mean that, if we see nothing where, in another paradigm, we would see something, we still believe that the same morphosyntactic information is represented in the mind. In Latin, for example, the noun *anser*, "goose," appears to have no suffix in the nominative, since all suffixes, like genitive *-is*, are concatenated with the stem, leaving *anser* intact in *anser-is*—while *cibus*, "food," *does* seem to have a nominative suffix, *-us*, because the genitive suffix *-i* replaces it to give *cib-i*. Similarly, the past tense of English *cut* is *cut*, while the past tense of *slice* is *slice-d*, not *slice*. We assume that these non-suffixes are really null suffixes, which is to say that the relevant morphosyntactic information is present no matter what. Other theories of morphology might lead us to different interpretations of what it means to be a null suffix. We assume null suffixation, and implement it as a restriction on concatenation—exactly one past tense suffix must be used on a past tense form, and the suffix must thus be null when there is no change. On the other hand, zero or more internal changes can apply to a stem, and we do not, therefore, generally have a use for null internal changes.

SPE-style environments have the general form in (15).

(15) X_Y

Associating such an environment with some phonological process means that that process targets segments exactly when they are between instances of the sequences of feature complexes *X* and *Y* (exactly those sequences, or more specific ones). SPE-style environments (sometimes the word *rules* is used to mean any rule with this kind of environment) have sometimes been contrasted with *analogy* (Pinker 1999, for example, calls the environments in which speakers will extend the *win–won* pattern "not very rule-like" to distinguish them from SPE-style environments). What exactly the supposed distinction is is hard to pin down, and will be discussed at greater length in Chapter 2.

The assumption of *ordered rules* means that speakers' knowledge of the mapping between stored forms and surface forms takes the form of a collection of smaller processes (each with an SPE-style environment, we assume), ordered either disjunctively (the highest-ordered process that matches applies, as we assume to be the case for suffixes) or conjunctively (each process that matches applies in turn to the output of the previous process). An ordering is necessary for disjunctive mappings if the environments for two different processes match some form but only one is correct. The linguist committed to a no-ordering theory could easily reformulate one of the environments to exclude the relevant form at the cost of having more complex rules, but we will attempt to build a learner that does not do this; it will thus need some ordering to compensate. Crucial orderings in conjunctive mappings ("opacity") will prove to be mostly irrelevant to the internal changes we will be discussing, namely, the simple stem alternations found in the English past tense (*ring–rang*), but, since we we will assume ordering for suffixes, we will assume ordering for internal changes for consistency.

Our assumption of *ordered mechanisms* is that the output of concatenation (at least, of the concatenation of two morphemes, the most we will be dealing with) is not visible to the internal change mechanism (at least, not as it applies to changes in the already-concatenated morphemes)—whereas the output of the internal-change mechanism *is* visible to the concatenation mechanism (at least, as it applies to the internally-changed morphemes). We can imagine that the alternate possibilities would lead the learner to different kinds of grammars, but it is not immediately obvious that they would be wrong. We make this assumption mostly for concreteness.[8]

Finally, *stem-marking* supposes that forms following patterns that cannot be stated using phonological environments are generated online by rules applying only to (or not applying to) explicitly listed (that is, specially marked) sets of morphemes. This contrasts, again, with output-memorization, which in this case supposes that, despite whatever patterns speakers may show awareness of when new forms are elicited, if these patterns would need idiosyncratic rules, then the existing forms, unlike any novel forms speakers might generate, are not derived online by rule, but simply memorized. We will not assume output-memorization here either, and we will attempt in Chapter 2 to dispense with claims to the effect that output-memorization is *necessary* for such patterns.

This collection of assumptions is somewhat arbitrary, but not entirely so. It is chosen partly because a theory of learning morphology under these assumptions is expected to be more informative than a theory of learning morphology would be under alternate assumptions. Assuming full and uniform decomposition means that we cannot avoid saying something about morpheme segmentation (including segmentation with null morphemes), which is an unsolved problem as long as we assume any morphological decomposition at all. Research into the acquisition of SPE-style environments presupposes that arbitrary, constructed "environments" form a part of grammar; constraint-based phonological grammars deny this to a certain extent, and, as such, research into SPE-style environment acquisition can be seen as competing with research into the acquisition of constraint-based phonological grammars. Constraint-based grammars, however, cannot do entirely without arbitrary phonological environments, if only to deal with arbitrary morphophonology; the current work can thus also be treated as a complement to constraint-based-grammar acquisition research. The same could be said of research into the acquisition of ordered rules and ordered mechanisms: to the extent that modern, constraint-based theories of phonology are monostratal, the current work can be seen as challenging those

[8] Although there is some intuition behind it: the reason I qualify the statements about what changes are visible to what mechanisms is that we can imagine both of these mechanisms applying repeatedly if morphemes are concatenated cyclically. That is the intuition underpinning this assumption: affixation takes place cyclically, so that there will be some rules will be applied before affixation, to each of the pair of morphemes being concatenated, and then some rules will apply to the newly concatenated object, along with anything with which it is now combining, and so on. Under this view, the current assumption can be stated somewhat differently: we are modeling only the first cycle.

theories; to the extent that our theory of *some* part of grammar is crucially derivational, lessons from acquisition research assuming ordered rules and ordered mechanisms are almost certainly relevant. Finally, acquiring stem-marking for idiosyncratic processes is a kind of (simple) morphosyntactic feature learning with obvious parallels to the acquisition of gender/noun class information, and perhaps to morphosyntactic learning more generally. The current assumptions thus seem to be interesting ones. Although I will not attempt to address any of these side issues directly, I feel it is nevertheless advantageous to pursue a theory which is relevant to other problems.

In this chapter I have laid out the goals of the paper: constructing a learner for certain kinds of theoretically interesting morphophonological facts, with an eye to broader issues in language acquisition. The remainder of the paper is as follows. Before discussing the learner proper, I take a chapter (Chapter 2) to defend some of the theoretical assumptions I have just laid out for morphophonology against criticisms that they are unworkable. For completeness, however, I will also examine an empirical argument that these same assumptions are not only workable, but are necessarily correct, and find it to be similarly lacking. Readers uninterested in the debates surrounding the implementation of morphology can skip Chapter 2. Leaving the (I will claim, mostly uninformative) debates behind, I will review the details of the construction of the learner in Chapter 3. Readers interested only in results can skip directly to Chapter 4, which will report the learner's results on the English past tense data.

Chapter 2

Morphology and Morphological Processing

In Chapter 1 (section 1.4), I outlined the assumptions behind the current study. Two of these assumptions in particular have received a good deal of attention in the psycholinguistic literature: the assumption of *SPE-style environments* and the assumption of *stem-marking*.

Bybee and Moder (1983) report the results of a test of English speakers' behaviour generating the past tenses of nonce verbs. Their results are, they claim, consistent with Bybee and Slobin's (1982) hypothesis that the morphological generalizations speakers keep about how the past tense is formed are "not in the form of rules ... which rigidly specif[y] what can and cannot occur." Instead, they suppose that these patterns are stored as *schemas* similar to the one shown in (16), supposedly associated with verbs which have past tenses in [ʌ] (like *string* and *spring*).

(16) s*CC* ɪ ŋ

Bybee and Moder's schemas are statements of a kind of hypothetical average member of a class—in this case, the class of verbs which have past tenses in [ʌ]. Instead of specifying all the characteristics a class-member *must* have, as an SPE-style pattern environment would, schemas specify the characteristics a class-member *could* have. If a form has more of these characteristics, it will be a better member of the class, in the sense that speakers will be more likely to accept it or produce it; if it has fewer, it will be a worse member of the class.

The experimental results seem to bear out this hypothesis: monosyllabic nonce verbs with final velar nasals are likely to change their vowel to [ʌ], but this is not impossible for verbs *without* final velar nasals, just less likely. The authors also suggest that having a final consonant that is *either* a velar *or* a nasal seems to help. Verbs with larger onsets are also more likely to take the alternation than verbs with smaller onsets—which also follows the schema theory, because ***spl****ing* matches more of the schema than ***p****ing*—but speakers still do not find [ʌ] forms completely illicit just because they lack large onsets—and it helps a form to have a final [ŋ].[1]

Because an SPE-style environment could only report *yes*, if a form matched it, or *no*, if a form did not match, with no notion of how "well" a form fits, Bybee and Moder suggest that, at least for these patterns, SPE-style environments are not sufficient to account for speakers' behavior, and that we need to appeal to a distinct mechanism: schemas.

Bybee and Moder's schema proposal is not the only line of research to claim that (some) patterns are not handled by SPE-style environments. Rumelhart and McClelland (1986) famously interpreted their network model of English inflection "without recourse to the notion of a 'rule'," and other connectionist researchers have followed. On the other hand, the *dual-mechanism theory*, which I discuss in this chapter (Pinker and Prince 1988, Pinker 1991, Marcus *et al.* 1992, Pinker and Prince 1994, Pinker 1999), does make use of "rules," but contends that, when marked forms are associated with patterns that do not seem to be easily stated as simple phonological environments (the cases under discussion), there are no rules, and the forms are simply memorized. This theory, too, appeals to an alternative to SPE-style environments—some kind of (unspecified) "fuzzy" mechanism. The nature of proposed alternatives to SPE-style environments has been diverse (where they have been precise enough to judge), but the generic term *analogy* has sometimes been used to refer to such alternate theories of phonological patterning. I will use this term here.

Alternatives to SPE-style environments (analogy) have often been associated with the denial of one of our other assumptions: *stem-marking*. Recall that this hypothesis claims that morphophonological alternations are derived by rules, even when those rules are imperfect; that the language faculty has the capacity to mark certain rules—rules which do not apply across the board—as requiring stems to be specially marked. This would mean, for example, that the alternation [aj] → [ej] found in the past tense of *lie*, "recline" (*lay*) is processed online by applying a rule to the stem; but this rule would need to take stem identity into account in order to avoid applying to *lie*, "prevaricate," which has past tense *lied*—hence, the first sense of *lie*

[1] The vowel, they claim, has little effect—but this is perhaps because they assumed [ɪ] was the optimal vowel. Their data actually does show an effect of the vowel—nonce forms with the vowel [æ] are better than ones with [ɪ].

needs to be marked for the rule, and the rule needs to be a special kind of rule—a *morpholexical rule*—which only applies to marked stems. This is the stem-marking hypothesis.

Analogy is often associated with a competing theory: *output-memorization*. Output-memorization runs counter to any theory that some form is derived online, as the past-tense of an English verb might be—derived from the stem. It claims that the supposed online procedure is actually just a search through lexical memory. In the case of *lie–lay*, for example, such a theory would say that speakers do not carry out a vowel alternation to make the past tense; rather, they remember the entire past tense form—including the fact that it begins with [l], although they already have this information contained in the stem. If we believe that output-memorization is used (at a minimum) for forms which might in some other theory be handled by stem-marking (morpholexical patterns), then this is a denial of stem-marking, because there is no need for a rule at all when the entire output is known; the memorization-and-search mechanism is taken to be crucially different from a rule mechanism in some way.

This theory has come to be associated with analogy, to the point that the two are sometimes not even distinguished: researchers endorsing the dual-mechanism hypothesis, the theory, as I will discuss in this chapter, that the mind uses output-memorization for morpholexical patterns, but derives other outputs online, have often made arguments for this kind of output-memorization on the basis of the apparent existence of analogy (and Yang's (2002) argument against the output-memorization of the dual-mechanism theory, in favour of stem-marking, was described as pitting "rules versus analogy"). The output-memorization theory and the analogy theory are often seen as the same thing—but they are not.

There is a reason for seeing such an association: the output-memorization theory must explain why it is that, if speakers do not have *rules* that derive *stung* from *sting* online, they can nevertheless fit new forms, like *spling*, to the same pattern (*splung*). A theory like Bybee and Slobin's schema theory seems perfectly suited to fill such a gap: if schemas only give speakers a gradient sense of what is a better or a worse form, speakers cannot be certain of the [ʌ] past tense of *dig*—surely a far-out member of the class according to the schema—or even *sting*: *sting* is a good match to the schema, but even the best nonce matches to the schema evoke the [ʌ] past-tense response only about half the time. Because this kind of analogy theory needs another mechanism to account for speakers' certainty about existing forms, and the output-marking theory needs an extra mechanism to account for productivity, the two fit together nicely.

Nevertheless, the two hypotheses are independent, and are in no way the same. There is nothing about an analogy theory that would prevent stem-marking: why can't a schema be used to produce forms online, and, as such, sometimes require markings to operate? Indeed, we might interpret a network model of English inflection which derives inflected forms online on the basis both of a stem's phonology and of its semantics, while at the same time denying that its output can be interpreted as a set of rules with SPE-style environments, (for example, Joanisse and Seidenberg 1999), as just such a model. On the other hand, there is nothing about output-memorization—whether restricted to morpholexical patterns, as in the dual-mechanism theory or taken as holding across the board, as in so-called "exemplar" theories of morphophonology (like Bybee 2006)—that would rule out SPE-style environments as a theory of linguistic patterning; if results like those obtained by Bybee and Moder point to an analogy theory, it is because of the behaviour of speakers, not because they have memorized the outputs.

Besides, the claim that the kind of behaviour shown by speakers in the experiment by Bybee and Moder *cannot* be described by SPE-style environments is too strong, as has been shown recently by Albright and Hayes (Albright 2002, Albright and Hayes 2003). If we allow for a single change to be associated with multiple rule environments, and if we allow for sufficiently powerful rule environments, then the effect that Bybee and Moder associated with the schema in (16), for example, whereby speakers judge forms to be better class members in proportion to how many parts of the schema they match, can just as easily be associated with the set of rules in (17):[2]

(17)
$$\begin{array}{c} \text{ɪ} \rightarrow \text{ʌ} \,/\, \#CCC\_ \\ \text{ɪ} \rightarrow \text{ʌ} \,/\, CC\_ \\ \text{ɪ} \rightarrow \text{ʌ} \,/\, C\_ \\ \text{ɪ} \rightarrow \text{ʌ} \,/\, \_[\text{nasal}] \\ \text{ɪ} \rightarrow \text{ʌ} \,/\, \_[\text{velar}] \\ V \rightarrow \text{ʌ} \,/\, \_[\text{nasal}] \end{array}$$

The rules in (17) clearly use SPE-style environments; if we can get them to account for speakers' gradient judgments, then the conclusion of Bybee and Moder is indeed too strong—but, of course, these rules need *some* augmentation in order to account for the facts: traditional grammars employing SPE-style environments are not designed to produce gradient output. Albright

[2] Note that the results presented in Bybee and Moder 1983 are fairly coarse, and do not actually show all the subtleties the schema would predict—there is no clear contrast in judgments between words lacking *s* but beginning with clusters and words lacking *s* but beginning with single consonants, for example. See also fn 1.

and Hayes thus propose a different sort of grammar, which keeps track of various statistics about the "reliability" of a change in one of its environments, in order to assign higher ratings to forms generated by more reliable rules. The details are not important here, but the existence of such a theory means that no radical break from SPE-style environments is needed to account for the facts.

I will not use Albright and Hayes's procedure, because it seems to presuppose output-memorization (note that this further shows the independence of analogy and output-memorization); another way of extracting judgments from these rules might be for speakers to count the number of environments that match a given form, and use this to determine how good a possible output is. If the set of forms allowed by one environment is a subset of the set of forms allowed by some second environment, any form matching the first environment will match two environments, rather than one. Here, a form terminating in either a velar or a nasal would match one rule, but a form terminating in a velar nasal would match two, and thus be better; similarly for onset size: a form with two consonants in the onset would be good—matching two environments—but a form with three consonants would match three environments, and thus be preferred. I will explore this proposal in Chapter 4.

In this chapter, I will review some of the literature on the topic of output-memorization and analogy. In section 2.1, I will discuss some of the evidence claimed to support the dual-mechanism hypothesis which does not rest on the rule/analogy distinction, and try to determine whether a stem-marking theory could handle the same facts. In section 2.2, I will discuss an argument from the literature against analogy/stem-marking, and show that the facts do not allow such a strong conclusion when examined more carefully, suggesting that a more sophisticated approach will be needed to examine these questions properly.

2.1 The Dual-Mechanism Hypothesis: A Response to Ullman *et al.* (2005)

The *dual-mechanism hypothesis* claims that output-memorization is responsible for speakers' knowledge of forms inflected according to patterns that have exceptions, while forms that follow exceptionless patterns are derived online. Above, we discussed why, despite attempts to strongly associate morpholexical output-memorization with a pattern mechanism distinct from SPE-style environments (which I called by the general term *analogy* above: Pinker 1991, Pinker and Prince 1994, Pinker 1999; also Bybee and Slobin 1982), there is no necessary relation between these two ideas. Nevertheless, there have been several studies claiming to support the dual-mechanism hypothesis on grounds quite independent of analogy. If these results are correct, then there may be a problem with our assumption of stem-marking. In this section, I discuss the largest and most recent of these studies, a study of brain-damaged patients by Ullman *et al.* (2005).

This study bears on whether there is a dissociation between two different systems, one for exceptionless patterns (the English weak past tense), and one for morpholexical patterns (the English strong past tenses). Typically, in such studies, exceptionless and morpholexical past-tense processing are compared and shown to rely on different regions of the brain—with certain regions being associated exclusively with the exceptionless patterns, and certain regions being associated exclusively with the morpholexical ones—and, from this, we are to draw the conclusion that the two types of morphology rely on distinct mechanisms.

When examining any such study, however, we must be aware of the fact that, although it would represent real progress to be able to match higher-level "mechanisms" with brain regions, the conclusion that there exist two fully distinct, clearly definable mechanisms based on a neural dissociation is much too strong. An ERP study by Bartke *et al.* (2005) highlights this kind of misstep. This study was of normal speakers' activation patterns for German plurals. The German plural system is notorious for having numerous patterns that are not exceptionless. The only form usually assessed as being the exceptionless one is *-s*, a somewhat rare form, but behavioural data in the same study is taken by the authors to support the standard conclusion that the *-e* plural is some kind of intermediate case—neither fully exceptionless nor fully morpholexical—we might call it *exceptionful*. The dual-mechanism theory says nothing about intermediate levels of exceptionlessness.

Nevertheless, two different non-exceptionless plurals, *-e* (apparently an exceptionful plural) and *-en* (apparently a morpholexical plural), show evoked responses in rather different regions. The authors, somewhat unsatisfyingly, suggest that the data call for a *three* mechanism model, which they call the "extended dual mechanism model," with three separate mechanisms for exceptionless, morpholexical, and intermediate morphological patterns—but this could in principle be just about *any* theory, not just one where these three mechanisms are totally unlike each other. One gets the feeling that we could multiply "mechanisms" indefinitely by examining low-level neurology, without learning very much about morphological processing.

It is thus difficult to know what such results mean. Furthermore, there are many things different about the *-s*, *-e*, and *-en* suffixes, some having nothing to do with how exceptionless they are (surely their frequency, and, of course, their phonology); granting, however, that the crucial difference is in how exceptionless the suffixes are, we still cannot draw conclusions about just what *sort* of differences there are.

Nevertheless, if we have a theory of morphological processing, and it does *not* predict any obvious difference between two kinds of forms, and then we see such an unpredicted difference, we ought to defer to a theory that does predict that difference. If we imagine what we might find in the brain during the processing of morpholexical forms under a stem-marking theory, for example, it is not unreasonable to suppose that we would find that these tapped a special pool of resources not tapped by non-morpholexical items. Many studies of brain-damaged patients have been taken to show just this point (Ullman *et al.* 1997, Patterson *et al.* 2001, Bird *et al.* 2003, Ullman *et al.* 2005; see Embick and Marantz 2005 for discussion): speakers with damage to certain regions (regions associated with "fluent" aphasia) do worse on morpholexical patterns than on exceptionless ones, but speakers without damage to these regions, including speakers with damage to other regions (those associated with "non-fluent" aphasia), do not show this pattern. This makes sense if damage to these regions compromises the ability to access or make use of stem markings. and it also makes sense on the dual-mechanism view.

On the other hand, stem-marking *per se* gives us no obvious reason to predict the existence of regions tapped exclusively, or more heavily, by exceptionless pattern processing than by morpholexical processing—but the dual-mechanism theory does make such a prediction. If exceptionless marking is done by transforming the stem, but morpholexical marking is done by searching for a new stem in memory as the dual-mechanism theory claims, then damage which is largely to the transformation mechanism (if this is possible) ought to affect exceptionless marking disproportionately. We do not get this prediction from a stem-marking theory. This is not to say that there might not be some other theory compatible with stem marking that could predict such an effect, but predicting such an effect—but in the absence of such a theory, the existence of brain damage causing difficulty *only* in exceptionless patterns should make us prefer the dual-mechanism hypothesis.

Just such a dissociation is sought in Ullman *et al.* 2005, a report of the performance of several brain-damaged patients on English past-tense marking in three tasks: production (elicitation of past-tense forms to complete an oral sentence, given the stem earlier on in the sentence), reading (subjects were asked to read inflected past-tense forms), and judgment (speakers were asked to determine whether a given past-tense form was correct). The authors claim to find evidence of better performance by non-fluent aphasics on strong than weak past tenses in all three studies.

If this were true, it would be suggestive of the dual-mechanism hypothesis. In the production task, however, there is a problem to be overcome before we can draw this conclusion, as pointed out by Bird *et al.* (2003). This study, like the study by Ullman *et al.*, showed that patients with damage to certain areas—very broadly in the same regions as the lesions of the speakers in the Ullman *et al.* study—did indeed perform worse on weak than strong past tenses, in production and reading tasks similar to those carried out by Ullman *et al.* However, in the production task,[3] the effect of verb type—morpholexical or not—disappeared in a subset of the materials matched for frequency, imageability, (as rated by normal speakers), and past-tense phonological shape. For example, the weak verb *train*, with past tense *trained* (shape CCVCC), was matched with the strong verb *grind*, with past tense *ground*, which has the same shape (CCVCC) and similar frequency and imageability ratings, for the purposes of the analysis. There was no difference across matched pair items, which strongly suggests that the effect is actually not of verb type, but rather of one or more of the three factors controlled for in the pairing.

In the production and judgment tasks, the analysis in the Ullman *et al.* (2005) study does factor out frequency, but not phonological shape or imageability; a significant effect of verb type is seen with and without this manipulation, but the effect is smaller when frequency is factored out. The Bird *et al.* result suggests that either phonological shape or imageability might be responsible for the effect, rather than verb type—indeed, the conclusion is almost inescapable for the production task, where both studies have comparable data. [4]

As for the judgment task, there is no comparable paired data presented in Bird *et al.* 2003, but we should assume that the same result—the effect disappearing after controlling for shape and frequency—would hold there too unless we have a convincing reason to believe that the failure of verb type to have any effect is a special property of production tasks. (That factoring out frequency results in smaller effects in the judgment task for Ullman *et al.* is also suggestive.) Given that at least one factor known to have an effect on this sort of result was left uncontrolled-

[3] Actually, in *both* of the production tasks these authors carried out—they did one in which they presented patients with a sentence frame, and one in which patients were to respond with the past tense immediately after hearing the stem.

[4] Ullman *et al.* claim that a phonological explanation is impossible because one of the patients did not make any strictly phonological errors—they look for errors like *keep–kep*—but they say nothing of the other speakers, and the facts discovered by Bird *et al.* are more important than this.

for, we should not be convinced here either.

Nevertheless, the report by Ullman *et al.* (2005) also contains the result of another task—a reading task—and there, the materials *were* paired for frequency and phonological shape. This had no effect on the result, which was positive: the same patients who appeared to do worse on weak verbs in the other tasks (whatever the reason) *still* did worse on weak verbs here, despite the manipulation.

Actually this result was not new: the reading task in the study by Bird *et al.* showed just the same thing, despite its controls for frequency, imageability, and phonological shape. Those authors, however, had an alternate explanation for this: strong past tenses, written out, tend to be more dissimilar from their stems than weak past tenses. Given that written weak past tenses always contain their stems, it is not surprising that speakers with brain damage should be more likely to read the stem rather than the full past tense—by far the most common error. On the other hand, strong past tenses do not contain their stems, and are thus less likely to show this error.

Responding to this possibility, Ullman *et al.* report the results of another test on some of the speakers: they note that patients had relatively good performance reading words like *everyone* and *plane*. This is unexpected, the authors claim, if speakers really had the sort of difficulty in reading that Bird *et al.* suggest: if speakers read *wait* instead of *waited* because they see the word *wait*, why don't they ready *every* (or *one*) for *everyone* and *plan* for *plane*?

This is not entirely convincing—there are many reasons we can imagine, not least of which is the fact that reading a past-tense plausibly involves doing a different kind of morphosyntactic composition than does reading *everyone*, and there is clearly no composition involved in reading *plane*—but it is possible. It leaves the fact that Bird *et al.* report *no* effect in the production task a mystery, however. Why is there a difference between the results on the reading task and the production task in the Bird *et al.*? Is a reading task generally harder, incurring more "cognitive load"? This is surely not a reasonable hypothesis, since it was the reading task that the *largest* number of non-fluent subjects in the Ullman *et al.* (2005) were able to perform in the first place—many had to be excluded from the other tasks. The Bird *et al.* explanation seems more complete on this account, and the study by Ullman *et al.* is at best inconclusive in light of their results.

For the purposes of the current work, I will take it that there is nothing in the Ullman *et al.* study—the most thorough study of its kind—that rules out a stem-marking theory. Given the discussion above dismissing the claims against SPE-style environments, this means we can proceed as planned—but is there strong evidence in *favour* of these assumptions? Yang (2002) claims that there is; in the final section, I will investigate this claim.

2.2 The Free-Rider Effect: Yang (2002) Revisited

Marcus *et al.* (1992) carried out a large study of English past tense acquisition based on data taken from the CHILDES database (MacWhinney 2000). They found, among other things, that the likelihood of a child incorrectly marking a strong verb with the weak *-d* ending—either in addition to (*ranged*) or instead of (*ringed*) the correct past tense form (*rang*)—correlated inversely with the verb's past tense frequency in the adult input.[5]

In an effort to explore the predictions of a learning model along the lines discussed in section 1.3, Yang (2002) analysed the data collected by Marcus *et al.* and concluded that children's overuse of the *-d* past tense also reveals another phenomenon, one which apparently suggests a model of morpholexical storage like the one we are assuming here. In particular, the frequency of a particular verb apparently affects not only children's correct past tense marking of that verb, but also their correct usage rate for *all the verbs that follow the same rule*. The past tense, of *fall*, for example, *fell*, is of middling frequency as strong past tenses go, but *forgot*, which is of *lower* frequency, shows a *lower* rate of erroneous *-d* marking. This is unexpected if only the frequency of the verb is determining how often it will be used correctly. The proposed explanation is that the other past tenses which follow the same rule as *forgot*—especially the extremely frequent *got*—are much more frequent overall than the words which are claimed to follow the same pattern as *fall* (*hold*, *come*, according to Yang—if this seems strange, see below), and this leads to a relative improvement in children's performance on *forgot*. This phenomenon is called the *free-rider effect* (FRE).[6] Yang argues for a theory of morphology

[5] Their suggested explanation followed the dual mechanism theory: non-*d* past tenses are morpholexical, and so their outputs are memorized as separate lexical entries. Supposing that memorizing a word takes repeated exposures, lexical access to the stored past tense form will fail if that past tense has not been heard enough times. By the Paninian principle, these words will then fall under the exceptionless *-d* pattern (and whether this comes out as singly-marked *ringed* or doubly-marked *ranged* will depend on whether the child is confused about what the stem is). Just as for the aphasia data presented in section 2.1, however, this explanation of the course of acquisition data can easily be restated under a stem-marking rather than an output-memorization theory of morpholexical knowledge. Note also that this theory predicts that the reason for errors like *worser* (attested, at least anecdotally) are also due to learners thinking that *worse* might mean *bad*. This is not impossible, but it cannot be the account of *worser* when it persists into adulthood (and my impression is that it sometimes does). Thanks to Elan Dresher for pointing this out.

[6] There is an alternate explanation in the case of *forget–get*: *forget* is actually decomposed into *for-* plus *get*, and will thus automatically take the same alternations as *get*. This would not be evidence of a general free-rider

like ours on the basis of this phenomenon.

The argument for this is a familiar one: one theory predicts the effect, while the other is silent. We can, supposedly, easily develop a model of how children acquire morpholexical knowledge along the lines of the Variational Learning model discussed in section 1.3 (in this case, a stochastic model of how children learn which verbs are associated with which strong patterns), without assuming anything like the dual-mechanism theory discussed above, and instead assuming *rules*. This learning model predicts the free-rider effect. The dual-mechanism hypothesis does not come with any predictive learning model, and so the rule-based theory should be preferred to the dual-mechanism theory.

The goal of this section will be to evaluate the empirical claim—does the proposed learning model actually predict FRE?—but it is also worth examining the logic of the argument. Consider the proposed learning model, shown in (18).[7]

(18) On processing a past tense verb form X with stem x, to determine whether rule R gives $X\ldots$

a. Stochastically choose a marking grammoid for x—either allowing or disallowing R's application..

b. Stochastically choose a grammoid telling us whether R does or does not apply at all (or whether it really "exists").

c. Apply/don't apply R to x (falling back on *-d* if we don't apply R):

(i) If the output matches X, reward our beliefs in both x's marking grammoid *and* R's existence grammoid.

(ii) If the output does not match X, penalize our beliefs in both x's marking grammoid *and* R's existence grammoid.

effect, then, but only evidence of a free-rider effect for morphologically related forms. It is quite difficult to find a clearer example of the effect, however—part of the reason I undertook the current study.

[7] In order to aid in our discussion of how this model is relevant to the set of assumptions laid out in section 1.4, I have rephrased step (18a) so that the stem-marking nature of the theory is clear. The original is formulated in terms of the inclusion or non-inclusion of x in a "list" of stems to which R applies, but this is an implementation detail—"marking" or not "marking" each stem as licensed for rule R clearly induces a list of stems for which R is licensed. I have stated step (18b) in terms of the "existence" of the rule—that is, whether or not the rule should be considered to be a legitimate part of the grammar being acquired—although it is not clear whether the intent was something subtly different from this. I have made this change to underscore the fact that it is the *general* applicability of R that is in question in step (18b)—that is, whether it *ever* applies—rather than just whether it applies to x, which would, of course, be the same as step (18a), and would not give the desired result. The original presentation can be found in Yang 2002.

This algorithm follows the general Variational Learning scheme introduced in section 1.3, and we assume that the weight updates in step (18c) are done in such a way that we can make the same kinds of predictions discussed there—that is, that a learner will adjust belief in grammoids in favour of correct ones sooner if it hears more counterexamples to the alternative. In this case, we suppose that a learner will come to believe in x's marking for R and in R's existence sooner if it hears more examples of R being used for x, and of R being used for any verb, respectively. By the logic of SGT, we can use the strength of the learner's belief in a particular grammoid at some time to predict the rate of production of forms which could be produced exclusively by a learner believing in that grammoid. Here, we assume that we can predict the rate at which learners will use the inappropriate weak (*-d*) form for some strong verb.

In this learning model, there are two places the learner could go wrong in its inflection of a verb, with the result that it would be inflected with *-d*. First, it might fail to believe sufficiently strongly in x's being marked for R; second, it might fail to believe sufficiently strongly in R's existence. If we assume the strength of the first belief to be proportional to the number of times the learner has encountered X, and the strength of the second to be proportional to the number of times the learner has encountered any past tense inflected by R, we predict FRE.

What theory of morphology, exactly, does this system support? Yang calls it a *rule-based model*—and although "rule-based" usually refers to models assuming what I call *SPE-style environments* (see section 1.4 above), there is nothing here that would rule out some (other) version of "analogy," though, of course, a distinction between SPE-style environments and analogy is meaningless unless we really know what "analogy" is supposed to mean (see the introduction to this chapter above)—but "analogy" in this sense does not seem to be what is really relevant here. Rather, the suggestion is that the system relies crucially on a stem-marking rather than an output-memorization theory of morpholexical knowledge.

This view makes sense because the procedure in (18) requires that there be a relation between the strength of the pattern that X follows (R) and the rate of correct production of X. Under an output-memorization view of morpholexical knowledge, speakers do not need any knowledge of a past-tense form's morpholexical pattern to produce it—they just memorize the past-tense form. Thus there is no necessary relation between how well the learner has acquired the pattern and how well the learner should do at producing forms (at least, forms it has heard) which follow that pattern.

The dual-mechanism literature contains proposed accounts both of FRE (see Marcus *et al.* 1992, in which reference was made to FRE; Yang's argument that FRE rules out a dual-mechanism theory is novel, of course) and of speakers' ability to extend morpholexical patterns to new forms (Prasada and Pinker 1993)—never more precise than references to the fact that network models along the lines of Rumelhart and McClelland 1986 can learn at least some of the relevant mapping. It is sufficient to point to existing models if the goal is merely to demonstrate that there are models that can find mappings among stored forms, but there is no specific reference in the network model literature, as far as I know, to anything like the FRE, that is to say, an effect of the overall frequency of a class on the rate of acquisition of its members, regardless of individual frequency. It would perhaps not be surprising to see this effect, but until it is concretely demonstrated, Yang's explicit model of FRE is a better explanation.

This is the logic of Yang's argument, but it has a major weakness: while the presentation contains many measurements of error rates and frequency, it contains no measurement of *FRE itself*—only discussion of isolated examples which are suggestive of FRE. As noted, however, FRE was discussed in Marcus *et al.* 1992. These authors measured the correlation, over all verbs *v*, between the rate at which children make past tense errors on *v* and the total frequency of all the other verbs in the same "family" as *v*, controlling for the frequency of *v* itself. Although these "families" were defined somewhat arbitrarily, the authors nonetheless measured a correlation between the two variables which was found to be statistically significantly different from zero (which would mean there was no correlation). Certain potential problems with the analysis in this study (discussed below) suggest an attempt to replicate the result would be worthwhile. This section, therefore, is an attempt at a new and better analysis of the error–frequency relations observable in CHILDES.

In section 2.2.1, I will draw out the quantitative predictions of a slightly more general version of the learning model in (18). In section 2.2.2, I will extract new error and frequency data and compile a list of possible strong verb classifications attested in the literature, to be tested to see if they show the free-rider effect (that is, the predictions made in section 2.2.1). If certain ones do, and certain do not, then, even though the learning model does not have independent justification, we may take the free-rider effect as evidence suggesting both the learning model and the given strong-verb classification(s). In section 2.2.3, however, I present the results of statistical tests indicating that the free-rider effect does not, in fact, seem to hold for any of the classifications described.

2.2.1 The Predictions of the Learning Model

In section 2.2.3 below, we will be testing the hypothesis that, for some particular classifications of English strong verbs, there is a statistically robust quantitative relation between class frequency and correct usage rate in addition to the relation between individual verb frequency and correct usage rate. Recall from section 1.3 the assumptions of *Stochastic Grammoid Theory* (SGT): before a linguistic lesson is learned, all the possible grammoids for that lesson are believed in by the learner to some degree, and the task of learning is to make these beliefs approach certainty. All other things being equal, the strength of these beliefs should be reflected in the rate of the learner's production of utterances characteristic of each grammoid. Yang's inflection-learning model, described above, is consistent with SGT, and predicts that belief strengths—and thus the learner's rates of production of various correct/incorrect forms—should be related to input frequencies. Below, we will attempt to test the predictions of this model about the relation between the two input frequencies (individual verb frequency and class frequency) and the learner's beliefs. To do that, we need to know what we are predicting and what we are measuring.

In general, the quantity that we can measure from the input representing the strength of some belief will be found by measuring the number of utterances a child makes that are compatible with only that belief, and taking this as a fraction of the total number of utterances for all the possible beliefs. In a simple case like the one in section 1.3, the beliefs we need to examine will be the learner's beliefs in each of the grammoids solving some linguistic lesson: we would find the utterances that clearly reveal use of one or another grammoid for some lesson (in section 1.3 we used the examples of $\pm$*pro*-drop and $\pm$V2), and measure what fraction show the use some particular grammoid (like $+$*pro*-drop or $-$V2).

In general, however, the beliefs of interest will be more complicated than a single grammoid. In Yang's inflection-learning model, there are two linguistic lessons to be learned—whether a verb v is marked for rule r, and whether r exists as a rule—and thus two different grammoid choices involved on the part of the learner. The probability of both of two independent events occurring—in this case, the learner choosing each of the two correct grammoids—is calculated by multiplying the probabilities of each of the two events.[8] Thus, the learner will choose the

[8] To deny that the learner's two choices are made independently of each other would be to assert that, for example, if the learner decided it believed in the marking of v for rule r, it would then be more/less likely to choose rule r. While this is plausible, I will take independence to be the null hypothesis, correct in the absence of any evidence against it.

two correct grammoids $\top_{v,r}$ (*true that verb v is marked for rule r*; as opposed to $\bot_{v,r}$, *false that verb v is marked for rule r*) and $\top_r$ (*true that rule r exists*; as opposed to $\bot_r$, *false that rule r exists*) with probability equal to the individual strengths of these two beliefs multiplied together, $B_t(\top_{v,r})B_t(\top_r)$. Following SGT, we assume that taking the total number of correctly marked past tenses as a fraction of the total number of attempts at the past tense of v will give $B_t(\top_{v,r})B_t(\top_r)$.

If we examine the learning model, we can derive a formula for $B_t(\top_{v,r})B_t(\top_r)$ in terms of input frequencies, (the relative frequency of v, and the sum of all the relative frequencies of all the verbs other than v that are inflected by r), just as we did in 1.3 and Appendix A for the single binary-valued parameter Variational Learning model—but it is an extremely unwieldy formula. More importantly, I am not aware of any straightforward way of testing the prediction given by that complex formula—but if we make a few simplifying assumptions (details of these assumptions can also be found in Appendix A), we get the formula in (19), where f_v is some measure of the probability of the learner hearing v on an arbitrary input, and F_v is the sum of such probabilities for all the verbs inflected by r that are not v (I discuss how I obtained these probabilities in the next section). This should give a rough prediction of $B_t(\top_{v,r})B_t(\top_r)$ in terms of input frequencies. Complete justification is in Appendix A.

(19) $$\log(B_t(\top_{v,r})B_t(\top_r))^{-1} \approx \beta_0 + \beta_1 \log f_v + \beta_2 \log F_v$$

This simplified equation will allow us to use a statistical model to see how robust the free-rider effect is—if it does exist—so that we can take into account the possibility that the appearance of the free-rider effect might just be due to noise in the data. Simplifying the equation into this form lets us approximate the values of the coefficients β_0, β_1, and β_2 that will put the observed data points closest to the plane $\beta_0 + \beta_1 \log f_v + \beta_2 \log F_v$. (This is the ordinary least-squares regression model, the familiar "line of best fit" technique—here, plane of best fit.) We can then determine, based on how much variability there is in the distance between the plane and the observed data, how probable it is that the same observed data might be seen in a world where the coefficients β_0, β_1, and β_2 were zero—the world of the null hypothesis, in which there is no such relation between the input frequencies and the child's beliefs. Assuming that the model in (19) is a sufficiently good approximation to the model's predictions, testing to see if β_2 could or could not plausibly be zero will tell us whether the predictions are really true.

Note that the model that gives us the (approximate) prediction that allows us to use the regression model to test it assumes that each verb is inflected by only one rule. In the model for past-tense inflection I will develop in this paper, I will assume that this is not correct. Rather, I will assume a model more like the one in (20).

(20) On processing a past tense verb *v*...

a. For each morpholexical rule *r*, randomly select a marking for *v*, either allowing or disallowing *r*'s application.

b. For each morpholexical rule *r*, randomly decide whether *r* does or does not apply at all (whether it really "exists").

c. For each morpholexical *r*, apply/don't apply *r* to *v* (falling back on *-d* as an affix if the last *r* which is a suffix has not applied, and making no stem change not licensed by some rule):

(i) If the output matches the attested *v*, reward our beliefs in both *v*'s marking and *r*'s existence.

(ii) If the output does not match the attested *v*, penalize our beliefs in both *v*'s marking and *r*'s existence.

In this model, a verb may be associated with *more than one morpholexical rule* in the past tense. This must be the case under the assumptions about morphology laid out in section 1.4. There I made the assumption that every past tense bore some suffix, so that, for example, *ring* would have a past-tense suffix -ø, since no material is added to form the past tense, *rang*. I consider internal changes like the one that changes [ɪ] to [æ] in *rang* to be necessarily distinct from suffixes, so that there would also be an ɪ → æ rule implicated in the formation of the past tense *rang*, a rule which would (or at least could) require a stem marking different from the one required by the -ø suffix, and which would have an existence grammoid $\top_{\text{ɪ}\to\text{æ}}$ different from the null suffix's existence grammoid $\top_{-\emptyset}$. That is precisely the kind of model shown in (20)—a simple generalization of the model given above—and it is the one that I will use below. There, I test a handful of different ways of classifying the English strong verbs into the patterns/rules they are associated with, to see if any show the free-rider effect. For those classifications that assume a single rule for each verb (most do), the model in (20) is exactly equivalent to the simpler model, so our prediction about the model's behaviour needs no change. For classifications that make use of multiple rules (there is one), we need a more complex prediction.

We might think that, because the model in (20) is a simple generalization of the model from which we derived the approximate prediction in (19), we could find just a simple generalization of our earlier quantitative prediction—and indeed, this is precisely what I will do—but it must be acknowledged that this will make the simplified prediction deviate even further from the real predictions of the model.

Here is why: one immediate prediction of the model in (20) is that, for verbs inflected by more than one rule, children should make all combinations of errors attributable to missing markings. Supposing a past-tense verb is inflected by one suffix s and exactly one internal change rule i, we should get errors like *ring–ringed* if the child fails to mark the verb for the rule, or to assert the existence of the rule, *for both rules s* and *i*; errors like *ring–ranged* if the child fails to mark the verb for s, or to assert the existence of s, but gets i right; and errors like *ring–ring* if the child fails to mark the verb for i, or to assert the existence of i, but gets s right. In general, for some verb v that is inflected by the rules $\{r_1, \ldots, r_n\}$, the quantity that will tell us how often the child will not make any of these kinds of errors is the probability of the child selecting all the correct markings and all the existence beliefs for all the rules—again given that all these choices are made independently of each other, this is $\prod_{i=1}^{n}(B_t(\top_{v,r_i}) \times B_t(\top_{r_i}))$, that is, the quantity obtained by multiplying all the $B_t(\top_{v,r_i}) \times B_t(\top_{r_i})$'s together.

The available data does not give us this quantity exactly, but it is close. The data I use below comes from Marcus *et al.* 1992; in that study, only the first two kinds of errors—like *ring–ringed* and *ring–ranged*—were included, while the number of *ring–ring* type errors were not counted. The data gives the ratio of the total number of correct markings[9] to what the authors consider to be the total number of attempts at the past tense—but, for them, attempts include only correct inflections plus *ring–ranged* errors and *ring–ringed* errors; *ring–ring* errors do not count as attempts at the past tense. (After all, one can easily imagine the difficulty we would have in identifying such errors in a thorough way!) A child's $\prod_{i=1}^{n} B_t(\top_{v,r_i})B_t(\top_{r_i})$ value, on the other hand, can only be properly estimated if we consider all the child's errors. This leads to the rate of correct usage being consistently underestimated by some small factor when we develop a prediction for $\prod_{i=1}^{n} B_t(\top_{v,r_i})B_t(\top_{r_i})$, but it is the best we can do without working out a much more complicated formula that would tell us the prediction of the model for the belief we are *really* able to measure—only to grossly simplify the formula again in order to test it with the regression model.

What *is* the prediction we will make for $\prod_{i=1}^{n} B_t(\top_{v,r_i})B_t(\top_{r_i})$? By the same law of diminishing returns that we used to avoid working out a precise formula for the quantity we are really ob-

[9] Actually, of incorrect markings, but we can easily compute the other ratio.

serving (which is not quite $\prod_{i=1}^{n} B_t(\top_{v,r_i})B_t(\top_{r_i})$), we will substitute a formula the predictions of which we can test using the regression model for one we know to be right. In particular, we will assume that the expansion of the formula for $\prod_{i=1}^{n} B_t(\top_{v,r_i})B_t(\top_{r_i})$ can be approximated by substituting, for F_v in (19), the product of *all* such terms, one for each rule r that inflects v (which we write $F_{r,v}$).

$$\log(B_t(\top_{v,r})B_t(\top_r))^{-1} \approx \beta_0+\beta_1\log f_v+\beta_2\log\prod_r F_{r.v} \tag{21}$$

The way we will use this approximation is as before: once we have the data, and have taken the appropriate transformations, we will find the best values of β_0, β_1, and β_2. We will then use standard techniques to test whether these values are statistically significantly different from zero.

Note that we have constructed this prediction on some arbitrary assumptions (see also Appendix A), but we are doing well comparatively. Working backwards from the discussion in Marcus *et al.* 1992, the test those authors carried out for their equivalent of FRE seems to have assumed, for no good reason that I can see, a relation between error rates and frequency which would translate into our theory as (22) (they assumed only one class per verb); and, as discussed above, Yang's (2002) assessment was based on qualitative inspection of the raw data only.

$$B_t(\top_{v,r})B_t(\top_r) = \beta_0+\beta_1\log(f_v)+\beta_2 F_v \tag{22}$$

I will spend the rest of this section evaluating the FRE claim assuming the model in (21). In the next subsection, I discuss my data collection; I believe the assumptions I make there about the correct treatment of the child data to be more reasonable to those taken up in previous studies, but anyone who might disagree can at least be satisfied that all my data is provided in Appendix B. I then report the results of the statistical tests, which are negative.

2.2.2 Compiling Data and Alternate Hypotheses

Having established the predictions of the learning model, we can now collect the relevant data. We will use the electronic CHILDES corpora of nine of the nineteen children studied in detail in Marcus *et al.* 1992:[10] Adam, Eve, Sarah (these three from Brown 1973), Abe (Kuczaj 1977), April (Higginson 1985), Naomi (Sachs 1983), Nat (Bohannon and Marquis 1977), Nathaniel (MacWhinney 2000), and Peter (Bloom *et al.* 1974). We will rely on some of the figures collected by Marcus *et al.*, but we will also need to collect some additional data from these corpora.

We begin with the data collected by Marcus *et al.* The authors counted the number of times several children made errors—either *ring–rang* or *ring–ranged* type errors—on a number of strong verbs, summed over the entire period represented by that child's corpus in CHILDES. Several things must be factored out of this data before it can be used. The first thing we must control for is the frequency of the past tense. If a child said *gotted* fifty times and *ringed* only once over a period of several months, it does not necessarily mean that the child had learned *ring–rang* better than *get–got*—there are also more opportunities to say the past tense of *get*, because *getting* simply comes up in conversation more often than *ringing*. For this reason, Marcus *et al.* present error *rates* rather than counts, the complement of which (that is, one minus which) gives *correct usage rates* (CUR), as shown in (23). We will start with these numbers, taken directly from Marcus *et al.*'s data by subtracting error rates from one.

(23) $$\frac{\text{Times some child got the past tense of some verb right}}{\text{Times that child attempted the past tense of that verb}}$$

Since this quantity is less informative the smaller its denominator (that is to say, if a child only attempts a particular past tense once and happens to get it wrong, should we care?) we will also follow Marcus *et al.* 1992 in only including data for a particular verb for children whose corpora show at least ten attempts at that verb in the past tense (verified by a manual search through the relevant corpora).

Furthermore, in this study we will use an aggregate measure of correct usage, taken across several children. If we did not do this, and instead analysed each child separately, we would be left with few data points (that is, few verbs), in each analysis, because each child can be seen to attempt a rather different subset of the strong verbs, and we want to have as many data points as possible to make sure that the free-rider effect is not an accident. Some care must be taken when aggregating the data. It would be wrong to change the calculation of CUR in (23) to one where we divide the number of errors on some verb *totalled over several children* by the total

[10] Marcus *et al.* studied Allison (Bloom 1973), but her corpus did not meet our threshhold of containing at least ten uses the past tense for any strong verb (see below). The remaining ten small corpora from Hall *et al.* 1984 contained very few verbs which met the ten-use criterion, and were not included.

number of times *any of those children* attempted to use that verb in the past tense, as in (24).

$$\text{(24)} \qquad \text{Overall CUR} = \frac{\text{Total number of correct responses from all children}}{\text{Total number of attempts by all children}}$$

If we used a measure like that, one child who used a verb far more frequently than any others would skew the overall rate for that verb unfairly. For each verb, therefore, we thus calculate individual children's CUR's and aggregate these by taking the arithmetic mean over the children who used that verb at least ten times in the past tense.[11]

$$\text{(25)} \qquad \text{Average CUR} = \frac{\text{Sum of all CUR's for any children with enough attempts}}{\text{Number of children with enough attempts}}$$

We must also collect adult frequency data. Because the raw frequencies of adult utterances of individual verbs in the past tense are not presented child-by-child in Marcus *et al.* 1992, I collected this data by manual search through all the relevant corpora. The only exception was *get–got*. This verb is very difficult to count, because instances of *got* are very often totally ambiguous, even in context, between a past tense meaning "received" and a present tense meaning "have" (which is presumably the reason that *got* has taken on this second meaning). It is also very frequent—so sorting out the present from the past instances presents a time-consuming challenge. This verb was thus extracted only for Adam, Eve, and Abe.

After extraction, the adult frequency data for each child was normalized to the number of adult utterances in their corpus, as shown in (26).

$$\text{(26)} \qquad \frac{\text{Number of adult uses of the correct past tense of some verb in some child's corpus}}{\text{Total number of utterances in that child's corpus}}$$

[11] Note that I do not make the move, as Marcus *et al.* do, of standardizing children's correct usage rates for each verb, so that they are expressed relative to how often that child overregularized overall. This makes the interpretation of the results far more difficult, and is of no apparent advantage. Furthermore, it reduces inter-child variation, which is bad, because it sets our standards lower—what we are attempting to do when we test for a relation of frequency to correct usage rate is testing an *explanation for the variability in the data*. If we remove variability unneccessarily, we can claim that we have a better explanation for the variability in the data than we really do, because there is less variability to begin with. Of course, removing variability is precisely what we are doing when we average children's overregularization rates; indeed, in the section of Marcus *et al.* 1992 addressing the free-rider effect (what they call *attraction to families of similar irregular verbs*), the authors do not take an average at all, claiming that the individual parent frequencies were too different to make an aggregate measure meaningful. This would lead to the sample size problem noted above, however. The frequency data I collected, which was taken relative to the total number of utterances in a given corpus, (see below) was *not* very different from child to child: the Pearson correlation from child to child had a median of 0.71, was never less than 0.48 (for April and Peter), and always t-tested significantly different from zero with a p value of no more than .00004 (for Naomi and Nat).

The normalized frequency was then aggregated, for each verb extracted, by taking an arithmetic mean over the children whose error data was being used for that verb (for *get–got*, of course, only over Adam, Eve, and Abe). For verbs not used at least ten times by any of the children, the frequency data for all nine children were aggregated, and this number was used to calculate class frequency. This was done because, even if we have no evidence about a child's performance on a given verb, by hypothesis, we still expect its frequency to contribute to that child's performance on other verbs. We have thus discussed all of the data used in the study; it is presented in full in Appendix B.

We now turn to the set of verb classifications we will test. Recall that we intend to see if at least *some* ways of classifying the English strong verbs into various patterns show the free-rider effect, since there is more than one way to classify the verbs. If no classification shows the effect, we should doubt its existence.

The classification of Bybee and Slobin 1982 divides the English strong verbs into eight classes, according to whether they undergo internal changes, what changes to final consonants or suffixation they show, whether they end in a consonant, and in particular whether they end in dentals; details are given in that paper. See Appendix C for a complete listing.

The classification of Yang 2002 works by sorting strong past-tenses according to rules needed to generate them: *-t* suffixation, *-d* suffixation, and -*ø* suffixation, along with a variety of internal changes (Shortening, Lowering, Backing, Umlaut, Vowel → u, and Rime → u). Because his model allows for a single rule for each verb, Yang treats each combination of internal change as a single rule, so that Shortening plus *-t* (*keep–kept*) is an entirely different rule from Shortening plus -*ø* (*meet–met*). While I do have a model that would allow us to separate the rules, I will keep this assumption to stay consistent with Yang's theory. Several of the verbs I used in this study were not in Yang's classification; I have classified these myself, and marked them in the full listing in Appendix C.

Marcus *et al.* 1992 classified the strong verbs in three different ways: a first classification, in which verbs were placed in the same class if they rhymed in both the past and the present (*sting–stung* and *cling–clung* would be in this class, but not *sing–sang*, because it does not rhyme with the other verbs in the past, or *hang–hung*, because it does not rhyme with them in the present); a second classification, in which verbs were placed in the same class if they shared a coda in the present and underwent the same alternation (*stick–stuck* would be in the same class as *sneak–snuck*, but *slink–slunk* would not be in this class, because the others do not have an [n] in the coda, although all three show the same change; *wake–woke* would also not be in this class, because it does not undergo the same alternation, although its coda, [k], does match); finally, a third classification, in which verbs were placed in the same class if they shared final consonants in the present and underwent the same alternation (*stick–stuck* and *sneak–snuck* would be in the same class as before, but now *slink–slunk* would also be in that class, because its final consonant is [k], and the preceding [n] is irrelevant).

Two of these classifications rely on the notion of "same alternation"; whether two verbs take the "same" alternation is not always evident, however: why is *sneak–snuck* the "same" alternation as *stick–stuck*, but not *wake–woke*? None takes *exactly* the same change. The authors make reference to the classification in Pinker and Prince 1988 as their source for the classification of alternations. However, it is not always clear how that classification was interpreted by Marcus *et al.*, because the classification in Pinker and Prince 1988 classifies changes in two ways: broadly (large Roman numerals in their text) and narrowly (Arabic numerals in their text). I include both the broad and the narrow classifications.

The rule system of Halle and Mohanan 1985 is in the generative tradition, and assumes stem-marking and SPE-style environments. In this scheme, each verb is generated by a derivation which requires that it be marked for several morpholexical rules. Recall that we developed a special model for testing theories in which a verb is inflected by several rules in section 2.2.1, which takes into account the frequencies of the other verbs inflected by the same rule, for each rule which inflects that verb. I will not discuss any details of this rather complicated system here; Appendix C gives references to the numbered examples in Halle and Mohanan 1985, q.v.

There is an extra complication when dealing with this theory, however: many of the underlying stem representations proposed by Halle and Mohanan are opaque—that is, they do not follow the assumption that the underlying form of that verb is the surface present/infinitive form. That assumption (which I will follow in this paper) makes the underlying representation of the past tense of *win*, *won,* /wɪn/, and derives the past tense from this. On the other hand, for Halle and Mohanan, the underlying representation of the stem *say* is /seː/, which (as innocuous as that might sound) is actually opaque: Halle and Mohanan assume a version of the synchronic vowel shift rule of Chomsky and Halle 1968, which means that /seː/ will actually surface as [si]! They assume that there is an internal change not only in the past tense of *say*, but also one in the present tense. This minimizes the number of rules.

Our simplifying assumption in section 2.2.1 (also Yang's assumption, and our assumption throughout this paper) was that children can immediately extract the correct underlying representations of all the past tenses they hear—and, *a fortiori*, that they know what the correct underlying representations are. This is a simplifying (that is, not entirely realistic) assumption to begin with, but when the correct underlying representations are claimed to be opaque, it becomes more unrealistic still: we cannot expect children's inflection of *said* to get a boost from other past-tenses inflected by the same rule if they have not yet learned that the underlying representation of the verb is really /seː/. For simplicity, however, I will ignore this extra complication.

These eight classifications are all tested in the following section. The classifications they yield are given in Appendix C. We proceed to the results.

2.2.3 Results and Discussion

Before doing any tests of the predicted relation from section 2.2.1, let us recall what the prediction is (in the general, multiple-rule case):

(27)
$$\log(B_t(\top_{v,r})B_t(\top_r))^{-1} \approx \beta_0 + \beta_1 \log f_v + \beta_2 \log F_v$$

A linear regression will find the values for β_0, β_1, and β_2 which minimize the overall distance between the data points and the plane. Our goal in this section is to examine the verb classifications in Appendix C to see if any will give us significant results for this model.

Table 2.1 demonstrates that none do. The F-statistic p-value indicates whether the plane is statistically significantly different from zero. The coefficients β_1 and β_2, and the intercept, β_0, also have p-values associated with them, indicating whether they are statistically significantly different from zero—important here since if all are not different from zero, we should doubt the free-rider effect. There are no classifications which show a statistically significant β_2. We thus find no evidence for the free-rider effect.

How could it be that we find no evidence of the effect apparently demonstrated in both and Marcus *et al.* 1992 and Yang 2002? The latter work does not present any statistical tests, and at any rate aggregates the CUR using the method we rejected in the previous section as being too sensitive to individual children. The former work reports the mean over 19 children of the partial correlation coefficient of family frequency for that child, holding individual verb frequency constant, to be significantly different from zero—that is, the data were not aggregated, which we argued they should be in the previous section, since otherwise each of the samples would

	F	β_0	β_1	β_2
Bybee and Slobin 1982	$p = 0.001$	[0.87 ($p = 0.06$)]	0.11 ($p < 0.001$)	[0.05 ($p = 0.50$)]
Yang 2002	$p = 0.001$	0.57 ($p < 0.05$)	0.11 ($p < 0.001$)	[-0.005 ($p = 0.71$)]
Marcus *et al.* 1992, Fam 1	$p = 0.001$	[0.54 ($p = 0.14$)]	0.11 ($p < 0.001$)	[-0.004 ($p = 0.78$)]
Marcus *et al.* 1992, Fam 2a	$p = 0.001$	[0.50 ($p = 0.09$)]	0.11 ($p < 0.001$)	[-0.008 ($p = 0.50$)]
Marcus *et al.* 1992, Fam 2b	$p = 0.001$	0.77 ($p < 0.05$)	0.12 ($p < 0.001$)	[0.009 ($p = 0.50$)]
Marcus *et al.* 1992, Fam 3a	$p = 0.001$	0.57 ($p < 0.05$)	0.11 ($p < 0.001$)	[-0.005 ($p = 0.69$)]
Marcus *et al.* 1992, Fam 3b	$p < 0.001$	0.84 ($p < 0.01$)	0.12 ($p < 0.001$)	[0.013 ($p = 0.31$)]
Halle and Mohanan 1985	$p = 0.03$	[0.20 ($p = 0.21$)]	0.05 ($p < 0.01$)	[-0.002 ($p = 0.85$)]

Table 2.1: Results of a regression under the approximate model. Figures enclosed in square brackets are not considered statistically significant.

contain very few data points. The fact that the authors were able to use all 19 children individually indicates that they attempted to alleviate this problem by temporarily letting go of the requirement that a child's error data for a given verb should only be included if the child used that verb at least ten times—but this move also leads to the individual partial correlations being unreliable, since most of that data is not very meaningful. It is thus important that we know which of these correlations are influencing the mean most heavily—if those represent less reliable measurements, we should doubt the result. I ran regressions using the relation assumed by Marcus *et al.*, on the individual data for Adam, Sarah, and Abe, the four corpora with the most data; none had coefficients of class frequency significantly different from zero, indicating that the mean of the partial correlations in the Marcus *et al.* study was probably skewed unduly by the less reliable correlations. We thus conclude that, despite its intuitive appeal, the free-rider effect does not seem to have any positive evidence in its favour.[12]

[12] Mayol (2003) reports an interesting fact about the acquisition of certain non-exceptionless Spanish verbal alternations which occur unpredictably in the first conjugation, more frequently in the second conjugation, and predictably in the second conjugation. Adult frequency and conjugation class, Mayol notes, are both good predictors (controlling for each other) of children's CURs. This is not necessarily the same kind of fact, however, since conjugation classes do not necessarily correspond to "rules" in Yang's sense. That is, whereas Yang's claim was that the data ran counter to the theory, attributed to Pinker, that rules are not used in the inflection of morpholexical patterns, it is not clear that the dual-mechanism theory requires that all arbitrary classes be handled by memorising outputs, although, if that is the case, it appears to be an inconsistency. Supposing that arbitrary marking of stems for conjugation class were possible under a dual-mechanism theory, then the alternation would be taken as the default for the third conjugation, and perhaps even for the second conjugation, and there would be no problem. This is granting a lot to the dual-mechanism hypothesis, however, and once the step of allowing stem-marked conjugation classes is made, it is hard to see why we would not allow stem-marked classes in English verbs.

Everything reviewed in this chapter is similarly inconclusive: the distinction between rules and "analogy" is not obviously meaningful; the sharp neural dissociation between exceptionless and morpholexical patterns is far from clear, the reasoning behind it dubious; and the impression we may get that the patterns children extract in morphology are reflected in their error rates either is obviously false or cannot be investigated in any good way. Where does this leave us?

It ought to leave the study of morphology more or less where it started: examining speakers' responses on judgment and elicitation tasks and attempting to construct formal models to account for them—the data from other domains, while interesting, are so far too unclear to extract anything meaningful about what a theory of morphology should look like.

It leaves the current work more or less anywhere we like—there is no convincing evidence from acquisition or neurolinguistics that we should reject rules and stem-marking (nor is there any knockdown evidence that we should endorse such a theory, of course). The long-term goal of a learner for morphophonology is to allow more evidence to be brought to bear on interesting issues like the abstractness of storage and the nature of generalizations: in order to make proper use of data the acquisition data, we need to construct explicit models of the acquisition process. This paper is a preliminary attempt towards that goal.

Chapter 3

Constructing a Learner

In this chapter I will describe a learning system for morphophonological alternations like those found in the English past tense. In section 3.1, I discuss how to construct an automatic learner using *Minimal Generalization Learning* (MGL; see section 1.2) under the assumption that suffixation and internal changes are distinct processes. In section 3.2, I present an extension to the system that implements the assumption of *stem-marking* discussed in section 1.4. Finally, in section 3.3, I discuss an extension to the system that keeps track of several versions of the same rule, in the spirit of Albright and Hayes 2003.

Throughout this chapter, I will present descriptions of the algorithms that make up the learner. I have decided to present each in informal prose for clarity, followed by a brief description of the intuition behind it—but I recognize that to make the procedures transparent it is not enough to avoid formalism. I have thus collected every component of the learner in Appendix D, along with at least one worked example of each. In the text, I will refer the reader to these examples as the components of the learner are presented.

Because some readers might be more or less interested in the details of the learner's operation, and the full motivation for these details, I have set off certain sections as minor sections by putting them in smaller type.

3.1 Learning Rules

Language learning is a search through the set of possible grammars, informed by the input, a steady stream of information about what the correct grammar is. A grammar, in turn, is a mapping between mental representations and utterances. As such, in order for a learner to get any useful information about its grammar, the learner's input must consist of more than just information about what acceptable utterances are; it must have information about acceptable *mappings* between mental representations and utterances. After hearing an utterance, the learner must guess at what the speaker's underlying mental representations were—semantic, syntactic, and phonological—in order to learn anything about how mental representations map to phonetics. In this paper, I will make the simplifying assumption that the learner has perfect knowledge of the speaker's mental representations, for reasons that are discussed in Chapter 1 (section 1.2).

How does a learner use its input to adjust its hypothesis about the correct mapping? A very general schema is given in (28).

(28) When told that s maps to t:

a. Use the current mapping, M, to find $M(s)$, our prediction of what s should map to.

b. Compute an *error signal*, some useful piece of information based on the difference between $M(s)$ and t.

c. Change M using the error signal.

Suppose that we are given a line to learn. We know that a line f is a mapping from real numbers to real numbers of the form $f(x) = ax + b$, and we are given as input a set of points on the line. Suppose we also know that a and b are integers.

To learn a line f, we would need some kind of initial hypothesis. We would then receive a steady stream of inputs, and we would need to do something with those inputs in order to adjust our hypothesis at each step. In particular, on each input $\langle x, y \rangle$, we would need to get the relevant information out of each input—that is, turn it into an error signal. Our general framework tells us to do this by finding out what our current version of f—call it $\hat{f}$—would give for x, and then comparing it to $\hat{f}(x)$ to y. This gives us a piece of information about what we need to do to make $\hat{f}$ correct.

There are a few different pieces of information that might be useful to us upon computing, that, for example, $\hat{f}(5) = 5$, when, in fact, we had just heard that the point $\langle 5, 10 \rangle$ was on the line $f(x)$. One useful possibility would be to find the difference between our prediction and the correct point ($10 - 5 = 5$). We could then do something with this information that would bring

$\hat{f}$ closer to f. For example, if we knew the difference between $\hat{f}(x)$ and $f(x)$, we could make changes by picking values of a or b that would make the error zero, though we would need to develop some criterion for determining *which* to change. We could also simply add one, positive or negative depending on the polarity of the error, to either a or b, or both. We could decide whether to change a or b or both by making a couple of extra predictions to see which would make $|f(5) - \hat{f}(5)|$ smaller: if changing a would make the error smallest, we would change a; if changing b would make the error smallest, we would change b; if changing both a and b would make the error smallest, we would change both. This would be a very general strategy that would be good for learning many sorts of functions, not just lines.

Suppose, however, the error signal was not the difference between $\hat{f}(x)$ and $f(x)$, but, rather, only the *polarity* of the difference between $\hat{f}(x)$ and $f(x)$—that is, a single piece of information telling us whether $\hat{f}(x)$ was above or below $f(x)$ at x. Then, although after each input we would want to change a and/or b, good strategies for deciding whether to change a or b or both would not be so obvious. We might decide to change a if our prediction was below the line, and b if it was above; or we might decide to change a if that would change the polarity of the error, b otherwise; or some other strategy—none seem to be obviously motivated. Nevertheless, we could likely find strategies that would eventually be effective, despite not having an obvious, general strategy.

How we change our model depends on what we get for our error signal, and the best error signal is one that is relevant to the sort of thing we are learning. When we are learning lines, we are searching through the set of pairs of integer parameters, a and b, for the pair that makes the error in our prediction—also an integer—go away. One can imagine that having an integer error signal would be generally fairly useful.

What about language? The type of language learning system we saw in Chapter 1 (section 1.3) uses an error signal that simply tells it whether the most recent prediction was correct or not, and it can quite handily (and successfully) adjust its confidence in the chosen grammoid using only that information. When we are learning sets of rules, however, unless we are learning them in some tricky, low-level encoding, we can expect that the most useful kind of error signal will probably be something like a rule. A number derived by some complicated formula, or a single piece of information telling us whether our prediction was correct or incorrect, will probably not do us much good—just as knowing whether the prediction is too high or too low is less useful than knowing *how far off the prediction is* when we are learning lines.

The *Minimal Generalization Learning* (MGL) strategy introduced in Chapter 1 (section 1.2) uses rules as its error signals: when it sees an alternation in its input (which is an underlying representation/surface form pair, $\langle s,t \rangle$), it finds a rule that would describe that alternation, and adds it to the grammar or generalizes an existing rule to include the environment of that alternation. The rule it finds is an error signal, in the sense that it is a useful piece of information about the difference between the current mapping—potentially one that would map s to s, not to t—and the correct mapping. By adding a rule mapping s to t, or extending an existing rule to do so, we make use of this information about how our grammar needs to change. MGL *per se* is an idea about *how* to incorporate such rule-error signals into a grammar—do it conservatively—but it could easily be incorporated into a fuller learning system.

The goal of this section is to develop a basic system that learns a grammar for the English past tense consisting of two sets of ordered rules (a set of internal changes, and a set of suffixes), according to the assumptions about the architecture of grammar made in Chapter 1 (section 1.4), using a system like the one in (29).

(29) On input $\langle s,t \rangle$:

a. Predict the past tense of s using the current rules.

b. Based on the attested output, t, and the predicted output, compute error signals that will tell the learner how it should change its grammar.

c. Change the grammar using the error signal, using the MGL strategy.

The outline of this section is as follows: first (section 3.1.1), I review the MGL strategy to clarify exactly how the error will be used to explore the set of possible grammars. I then (section 3.1.2) discuss how error is to be extracted, with further motivation presented in minor section 3.1.3. Finally, I present the results of simulation (section 3.1.4).

3.1.1 Minimal Generalization Learning

In the introduction to this section, I introduced the idea that learning takes place using *prediction* and *error*. This is a very general idea—so much so that it perhaps has no empirical force—but it gives us a general framework off of which we may hang our learning systems.

I also introduced the idea that, when we are learning a system of rules, we ought to have an error signal that gives us—more or less–*rules*. After hearing an input, we will suppose that our learner makes a prediction, and then examines that prediction to see what rule(s) would make its grammar make a better prediction, if any are needed.

In this section, I will explore a proposal, introduced in Chapter 1 (section 1.2) for how the learner uses these rules—its error signal—to update its grammar. This is a very simple proposal, based on the idea that phonological learners are *conservative*: they explore the most general rules last; they start with rules that are not generalized at all.

We begin with an illustration. Recall our assumption from Chapter 1 that learners (or, the parts of learning that we are studying here) take as input pairs $\langle s,t\rangle$, where t is the utterance heard by the learner (say, [ræŋ]), and s is the speaker's intended meaning, which, furthermore, contains all the necessary information about underlying forms (so s might be *Past tense of* RING, *which has underlying stem form* [rɪŋ]). Suppose that the learner makes a prediction about the past tense of RING and finds that its prediction is not [ræŋ], but, rather, [rɪŋ]. Surely the error signal should be something telling the past tense learner that [ɪ] should sometimes go to [æ]. Precisely how the error is to be extracted is a matter for another section (3.1.2 below). The MGL proposal is a proposal about what basic information the error signal should contain, and what should be done with it.

In this case, MGL would say that the error should contain the alternation (in this case ɪ → æ) and the full environment in which that alternation was attested (#r_ŋ#; I will assume that internal change rules can contain boundaries # in their environments)—for convenience we will sometimes simply state this as the rule ɪ → æ/#r_ŋ#—and that a *minimal* change should be made to the grammar to accommodate this new information.

If there is no rule ɪ → æ in the grammar, then the minimal change would surely be to add ɪ → æ/#r_ŋ# to the grammar. Adding this rule is as small a change to the system as is possible, because this is the rule that would attest for the observed alternation while affecting the fewest number of possible forms.[1] On the other hand, since we would like the system to learn something, rather than simply memorize its input, I will assume that adding a rule is impossible if there is already a rule ɪ → æ in the grammar. In this case, the minimal change will be to expand the scope of the existing rule. This is where *generalization* comes in.

Suppose that, after storing the rule ɪ → æ/#r_ŋ#, the learner gets a new input pair $\langle s,t\rangle$, where $s = \langle$SING, [sɪŋ]$\rangle$ (which I use to abbreviate *the past tense of* SING*, which has underlying form*

[1] We assume for the purposes of this section that the system does not allow for *stem-marking* for *morpholexical rules*, in the sense of Chapters 1 and 2. If it did, then we could add an even more restrictive rule—a rule ɪ → æ/#r_ŋ# that applied *only when the stem was* RING. We will treat stem-marking in section 3.2, though not using this strategy.

[*sɪŋ*]) and t = [sæŋ]. It then computes the error ɪ → æ/#s_ŋ#. Since the change ɪ → æ is already in the grammar, the learner will change its environment (as little as possible) to include #s_ŋ#, rather than adding this rule.

The details of this procedure for generalizing two rule environments are fairly straightforward: the learner separately compares the left environments (material that must precede [ɪ] for the change to be triggered) and the right environments (material that must follow [ɪ] for the change to be triggered) of old rule and new—comparing #r to #s and ŋ# to ŋ#. In each environment, it keeps as much identical material as possible adjacent to the change, until it encounters a non-identical segment. It collapses the first pair of non-identical segments using a feature system, and stops. The right environment, for example, would stay intact as ŋ#, because the system will run off the end of the two strings before it encounters any non-identical material. On the other hand, the left environment contains no identical material adjacent to the change—which is right of r/s—and so it would combine r and s using its feature system, and stop. Our feature system is given in Appendix E; it tells us that the left environment will be [+cons, +cont, +cor, −dor, −lab, −lat, −nas], the set of features shared between r and s.[2] There will be no # boundary, because the system stops when it reaches the first non-identical pair of segments closest to the change. The generalized environment is thus [+cons, +cont, +cor, −dor, −lab, −lat, −nas]_ŋ#. (A slightly more explicit version of this combination algorithm is given in (30) in the minor section below; a chart for following along with this example is given in (99). A second example is given in (100).)

This simple rule—generalize known environments to account for new forms, but generalize them as little as possible—is the basic idea behind MGL. The goal is to have a system that avoids overgeneralization as much as possible—and, indeed, it is largely successful (aside from the possible effects of morpholexical patterns and rule ordering—see Chapter 1, section 1.2), under certain assumptions about what rules are possible. See the minor section below for details.

This concludes our basic discussion of the MGL system, which we will use here to combine the rule information provided by the error signal with existing rules. Readers not in need of the slightly more explicit statement of the procedure combining two rule environments, or uninterested in the limitations of MGL's conservatism, can safely skip the following minor

[2] From now on I will remove features somewhat arbitrarily from representations if I find them to be redundant, unless it is crucial to understanding. The reader is expected to consult the feature chart in Appendix E as an aid if necessary.

section.

As discussed above, MGL will usually combine rules in a way that avoids overgeneralization, but it will fail to avoid it more often if we use more sophisticated rules. One entirely uncontroversial assumption that we must not make here for exactly this reason is the assumption that a change $A \rightarrow B$ can make reference to anything more general than specific segments. For example, we might want a change $[+\text{syll}, -\text{high}] \rightarrow \text{i}$ in certain environments, if we see two different non-high vowels changing to [i] in the same environment. Here we cannot do this. This has empirical consequences, as does any other statement of what possible generalizations there are.

Suppose that there were some collection of segments all undergoing some change—say, they change to [i]—all in the same environment; the only feature they share, according to our independently-motivated theory of the language's phonology, is, say, $[-\text{high}]$. Now suppose a speaker of this language encounters some sound which is also $[-\text{high}]$, but it has never encountered before in the relevant environment—perhaps it is a foreign sound, or a rare sound, or it only rarely occurs in the relevant environment. The theory disallowing the generalization of changes would then predict that the speaker would *not* change the sound to [i], despite its occurring in the appropriate environment; the theory allowing the generalization of changes predicts that the speaker would make the phonological change. It is widely accepted that the latter theory is correct, so our generalization system is incomplete. (We can make a similar argument about rules which have an underspecified feature complex on the *right-hand* side of the rule, as well—like $[+\text{syll}, -\text{high}] \rightarrow [+\text{low}]$—but I will leave these out of the following discussion for simplicity.)

Suppose we added this functionality to the learner. If the learner discovers that $\text{i} \rightarrow \text{e}/\text{p}\_$, and that $\text{i} \rightarrow \text{e}/\text{k}\_$, then, if [p] and [k] are the only sounds that are $[-\text{cont}, -\text{cor}, -\text{vc}]$, the learner has stated nothing new if it posits a rule $\text{i} \rightarrow \text{e}/[-\text{cont}, -\text{cor}, -\text{vc}]\_$. (If those are *not* the only sounds that are $[-\text{cont}, -\text{cor}, -\text{vc}]$, but these are the only features those two segments have in common, then we *have* stated something new, and so it is in principle possible, though hard to imagine, that our new generalization might be wrong.) Now, however, suppose that the learner discovers that $\text{a} \rightarrow \text{e}/\text{p}\_$. Again, if [i], [e] and [a] are the only sounds that are $[-\text{back}, -\text{cons}, +\text{syll}]$, then the learner will be asserting nothing new to state that $[-\text{back}, -\text{cons}, +\text{syll}] \rightarrow \text{e}/\text{p}\_$, because it is true that $\text{i} \rightarrow \text{e}/\text{p}\_$. If the learner *replaces* the rule $\text{i} \rightarrow \text{e}/\text{p}\_$ with $\text{i} \rightarrow \text{e}/[-\text{cont}, -\text{cor}, -\text{vc}]\_$ after hearing evidence that $\text{i} \rightarrow \text{e}/\text{k}\_$, however, can it now safely assert that $[-\text{back}, -\text{cons}, +\text{syll}] \rightarrow \text{e}/[-\text{cont}, -\text{cor}, -\text{vc}]\_$? Surely not terribly *safely*, since there is no evidence that [a] undergoes the shift after [k]—and hardly minimally, since there is another, still general way of stating these facts which would be more restrictive (namely, by not combining the [a] in $\text{a} \rightarrow \text{e}/\text{p}\_$).

In other words, given that two environments are associated with the same change, we are as certain as we can be that we want to generalize the two environments; on the other hand, given that two changes are associated with the same environment, we are just as certain that we want to combine those changes—but two changes associated with two *different* environments, we cannot be sure what to do with. There is thus no unique "minimal" generalization in that case, and the system breaks down. I leave this very basic problem for future research. (There are other sorts of rules that causes us to lack a unique minimal generalization, including rules using the C_0 notation or other ways of representing optional elements; thanks to Elan Dresher for pointing this out.)

A detailed and slightly more formal statement of the procedure for combining two rule environments is given in (30). The example from above can be found in the chart in (99), and a second example can be found in (100).

```
(30)  On input $\langle q\_r, q'\_r' \rangle$:
        1. There are two (possible empty) left environments to be combined, $q = q_m \ldots q_1$, $q' = q'_n \ldots q'_1$. There are two (possibly empty) accompanying right environments to be combined, $r = r_1 \ldots r_k$, $r' = r'_1 \ldots r'_l$.
        2. Let $q^g \leftarrow \varepsilon$, the empty string.
        3. For $i$ in $1, \ldots \min(m,n)$:
            (a) If $q_i = q'_i$, then let $q^g \leftarrow q_i q^g$.
            (b) If $q_i \neq q'_i$, then let $q^g \leftarrow Q_i q^g$, where $Q_i$ is the set of all features contained in both $q_i$ and $q'_i$. Stop looping.
        4. Let $r^g \leftarrow \varepsilon$, the empty string.
        5. For $i$ in $1, \ldots \min(k,l)$:
            (a) If $r_i = r'_i$, then let $r^g \leftarrow r^g r_i$.
            (b) If $r_i \neq r'_i$, then let $r^g \leftarrow r^g R_i$, where $R_i$ is the set of all features contained in both $r_i$ and $r'_i$. Stop looping.
        6. Return $q^g\_r^g$.
```

This concludes our discussion of MGL, the system we will use to interpret the error signal and update the rule systems, in the context of the current learner. I now turn to the procedure that finds the error signal in the first place.

3.1.2 Two Components, Two Error Signals

In the introduction to this section, I introduced the idea that learning a rule system could be thought of as taking place by making *predictions*, then getting back *error signals*. When learning a rule system, I argued, the error signals should be in the form of rules, or possibly some other piece of information that tells the learner how to adjust its set of rules more or less directly. In section 3.1.1, I discussed the procedure by which the learner adds or changes rules using basic error information, a simple conservative procedure I called MGL (Minimal Generalization Learning), following Albright and Hayes (1999). I did not discuss where this error information comes from.

In order to extract the kind of error signal we are looking for—a rule—from some pair of forms (like the prediction of our model and the input we just heard), we need simply to compare those two forms, and find where they deviate from each other. This will be the *change*. The remainder will be the *environment* of the rule. In this way, we extract rules that are specific to the form heard, and pass these on as errors to the MGL system, which takes care of constructing a grammar. We need to know how to extract these rules.

This topic might be obvious or uninteresting to some readers, and so I will present the majority of the detail here as a minor section. There is one thing, however, that is worth pointing out to those readers before they turn to the next section: one of our assumptions about morphology is that it has *two components*: internal change and suffixation. The internal change component consists of a set of ordered internal change rules, applying conjunctively to forms (that is, in sequence, with the output of one rule feeding the next rule); the suffixation component is a collection of suffix rules, applying disjunctively (that is, only one may apply—namely, whichever rule is the most specific match to the form) to the *output* of the internal change component.

This means something for error. As pointed out in the introduction to this section, error information must be appropriate to the sort of knowledge being updated. Internal changes are not suffixes, and so we cannot properly incorporate an internal change error into the suffixation component, nor a suffixation error into the internal change component. This suggests computing *two error signals*.

Furthermore, we will compute these error signals separately, taking into account that the two components need to interact. In particular, if the correct inflection of *flick*, [flɪk], is *flecked*, [flɛkt], but our model predicts [flɪk], then our suffixation error will be [t], the *-d* suffix, with an environment [flɛk], because our output did not have the right suffix, and [flɛk] is the environment in which the [t] suffix was seen to occur; our internal change error will be a rule taking ɪ → ɛ in the environment #fl_k#. If, on the other hand, the correct inflection of [flɪk] is [flɪkt], then, when our model predicts [flɪk], the suffixation error will be [t], as before—only this time with the environment [flɪk]—but there will be *no* internal change error. Although the final prediction deviates from the expected output, *the output of the internal change component was correct*—because it matches what the *input* to the suffixation component ought to be.

Readers who are still confused as to how this would work should continue reading this section (as, of course, should readers who are interested in the details). Readers who cannot imagine how *else* we would compute the error for the two components, or who *can* imagine it, and are therefore skeptical as to why this move is necessary, should read minor section 3.1.3.

We begin by considering the calculation of internal change error, for finding changes like the ɪ → æ change in $\langle$rɪŋ, ræŋ$\rangle$, and the i → ɛ change in $\langle$kip, kɛpt$\rangle$. Consider the straightforward procedure in (31).

```
(31)    On input $\langle u, v\rangle$:
        1. Let $m$ be the number of segments in $u$ and $n$ the number of segments in $v$. Then $u = u_1 \dots u_m$ and $v = v_1 \dots v_n$.
        2. Let $c \leftarrow 0$.
        3. For $i$ in $1, \dots, \max(m,n)$:
            (a) If $i \leq m$, $U_i \leftarrow u_i$; otherwise, let $U_i \leftarrow \perp$.
            (b) If $i \leq n$, $V_i \leftarrow v_i$; otherwise, let $V_i \leftarrow \perp$.
            (c) If $U_i \neq V_i$, $c \leftarrow i$. Break out of the loop.
        4. If $c = 0$, return $\emptyset \rightarrow \emptyset/\_$.
        5. Let $d_u \leftarrow m$; let $d_v \leftarrow n$.
        6. For $i$ in $c, \dots m$:
            (a) For $j$ in $c, \dots, n$:
                i. If $u_i = v_j$, let $d_u \leftarrow i - 1$, and let $d_v \leftarrow j - 1$. Break out of both loops.
        7. Let $A \leftarrow u_c \dots u_{d_u}$; let $B \leftarrow v_c \dots v_{d_v}$; let $C \leftarrow u_1 \dots u_{c-1}$; let $D \leftarrow u_{d_{u+1}} \dots u_m$; return $A \rightarrow B/C\_D$.
```

This procedure searches u and v left-to-right for the first point at which they deviate. It then searches, again left-to-right, for the next point at which they are the same (possibly different between u and v). It returns the change $a \rightarrow b$, where a is found in u and b in v at the point at which the two do not match. It returns the part of u that was not identified as being part of the change as the environment of that change. Multiple changes are not handled—additional changes that arise will be returned in the environment—but this does not cause problems in the English past tense. Some notable cases arise: when there is no difference, it returns the zero rule ø → ø, which we will take to cause the system to make no change; when the two strings are different lengths, it returns the extra material in the longer string as part of the change (so long as it is part of the *first* change). See (92)–(93) for examples.

This is about the simplest thing we could imagine for finding internal changes—line up the matched items and search through them until we find the point at which they differ—and it is sufficient here.

More important is the *input* to the procedure. As discussed above, we will base this error signal not on the difference between the final output of our derivation (our final prediction) and the attested surface form, but rather on the difference between the output of the internal change component and what *should* be the input to the suffixation component, given the attested surface form.

This is easy if we have a procedure for working out what suffix is to be found on the attested surface form; if, given *keep–kept* as input to the learner, we can always find the *-t* suffix, then we can always determine that *kep* is the target output for the internal change component. Furthermore, once we have found the suffix in the alternation, we can just compare it to the suffix that we actually applied (if any) to see whether we should send that suffix as an error signal to the suffixation component.

For both purposes—finding what the target input to the suffixation component should be, and finding the suffixation error—our job will be much easier if the suffix-extraction procedure always returns exactly *one* possible suffix, and operates independently of the internal-change error. To see what this means, consider the case of *bring* ([brɪŋ]), which has past-tense *brought* ([brɔt]).

How is this past-tense formed under our theory, which supposes that past-tense formation will be by a combination of suffixation and internal change? There are a number of possibilities. It might be formed by first applying an internal change ɪŋ → ɔt, then attaching the null suffix. It might, on the other hand, be formed by applying an internal change ɪŋ → ɔ, then attaching a suffix [t]. Finally, it might be formed by applying an internal change ɪŋ → ø, and then attaching a suffix [ɔt]. The most obvious cure for such uncertainty would be if the underlying form was entirely contained within the surface form, as it would be in *walk–walked*, ⟨wɑk, wɑkt⟩. There is only one possible suffix in this case, under any reasonable definition of "suffix." The input ⟨brɪŋ, brɔt⟩, however, does not have this property—and yet we want to find some kind of "suffix" in it.

This is easier in the case of *keep* ([kip]), which has past-tense *kept* ([kɛpt]). Here, the demarcation between internal change and suffix is much clearer. If we are going to divide this alternation into internal change and suffix, we will surely posit the internal change i → ɛ and the suffix [t]; it would be quite strange to think of this alternation as being the result of an internal change ip → ø and a suffix [ɛpt]. Why?

The general problem we are dealing with is has been addressed in the literature under the names *correction* or *alignment*, most notably by Wagner and Fisher (1974), an approach extended more recently in Kondrak 2002. These algorithms attempt to find the parts of two strings that correspond, by making a global evaluation of the possible sets of changes taking the first string to the second; they assign a cost each of the possible changes—insertions, deletions, substitutions, and some other basic changes—and then find which set of changes has the least overall cost.

Here, we assume that there are only two changes in any given alternation; once we know one, we know the other. There is always more than one possibility for each, so we need some kind of cost function to tell us which is best. We will then evaluate this cost function to find the least implausible internal change, or the least implausible suffix.[3]

Here we have already determined that we will find the suffix first, and then the internal change, since we are assuming that our internal-change error computation will derive an internal change taking the output of the internal-change component of the derivation to the final form *with the suffix already removed*, which presupposes that the suffix has been found. We must thus develop a cost function for suffixes.

In general, the cost of a suffix will depend on many things. Here we will make the simplifying assumption that it only depends on two, somewhat arbitrary facts about the pair $\langle u, v\rangle$, where we assume that v is the result of applying some suffix to u: first, the amount of material that is shared between the underlying and surface forms that is *immediately adjacent to the suffix*; suffixes which make more of the part of the stem at the right edge look unchanged between underlying and surface forms will be preferred.

For example, in ⟨wɑk, wɑkt⟩, if we choose the suffix -ø, then we can compare the three segments immediately adjacent to the suffix in the surface form, [ɑkt], to the three segments of u, [wɑk], the part of u that corresponds to the part of the surface form immediately adjacent to the suffix, because the right edge is where the suffix attaches to u. These strings are not identical; the largest length string we could adjacent to the suffix that would match the right edge of u would have length zero. On the other hand, if we choose the suffix [t], then the three segments adjacent to the suffix are [wɑk], *precisely* the three segments at the right edge of u. Thus [t] wins.

On the other hand, given ⟨brɪŋ, brɔt⟩, we would find that the longest string adjacent to the suffix that we

[3] We might instead evaluate a function of the *combination* of internal change plus suffix, in case there were some internal changes and suffixes that were perfectly good by themselves, but which together seemed implausible—but it is difficult to see why we would need to do this.

could ever make correspond to anything at the right edge of *u* would be of length zero—for consider all the possibilities:

- If the suffix were -ø, the adjacent string of length four, [brɔt], would not match the four segments at the right edge of *u*, [brɪŋ], nor would the the adjacent string of length three, [rɔt], match [rɪŋ], nor would [ɔt] match [ɪŋ], nor would [t] match [ŋ]; the longest string adjacent to the suffix matching the right edge of *u* would thus be the empty string.
- If the suffix were [t], the string of length three would adjacent to the suffix, [brɔ], would not match [rɪŋ], the three segments at the right edge of *u*, nor would [rɔ] match [ɪŋ], nor would [ɔ] match [ŋ].
- If the suffix were [ɔt], the string of length two adjacent to the suffix, [br], would not match the two segments at the right edge of *u*, [ɪŋ], nor would [r] match [ŋ], and we would also be left with only the zero-length string.
- If the suffix were [rɔt], the string of length one adjacent to the suffix, [b], would not match [ŋ]. Finally, if the suffix were [brɔt], the only string adjacent to the suffix would be of length zero to begin with.

All the possible suffixes given ⟨brɪŋ, brɔt⟩ share an equal cost according to this first criterion; we thus need a second criterion, equally arbitrary, but also simple—pick the shortest suffix. In this case, the shortest suffix is the null suffix, so we choose the null suffix for ⟨brɪŋ, brɔt⟩ (and that leads our internal-change error to be ɪŋ → ɔt/#br_#).

The main claim of this system is that suffixes are better—or, we might think, only identifiable—when what is adjacent to them does not change. Alternately, we could think of the system as claiming that, when there is a suffix attaching, *internal changes* should not be applying at the right edge of *u*. This is a good way of avoiding the possibility of having internal changes take over where we want suffixes—as in the case where we suppose that *walked* is formed from *walk* by applying a null suffix and inserting a [t]—but it can be taken too far.

Suppose we used our cost function to extract a suffix from the pair *understand–understood* (that is, ⟨ʌndərstænd, ʌndərstʊd⟩). We might think that the best suffix here would be the null suffix, because there are no possible suffixes for which any non-empty immediately adjacent material is to be found at the right edge of *u*, and we would then pick the null suffix because it is shortest. In fact, however, there is one suffix that fits the bill: *-erstood*. The two segments immediately preceding *-erstood* ([ərstʊd]) are [nd]. By what we can only assume to be a coincidence, these are also the last two segments of *understand*—but the learner does not know that this is a coincidence, and so it reports that the best suffix is *-erstood*. This is very wrong.

What it shows us is that the criteria for a good suffix, or a good internal change, are surprisingly far from being trivial. I will thus leave this empirical question open. As a quick fix to our cost function, I will make the learner immediately reject any suffixes that have length greater than three. This is not the right answer, but I do believe that there is something to the intuition that inflectional morphemes tend to be short, although this might be an accidental (e.g., historical) rather than a necessary property. This is summarized in (32).

(32) a. Suffixes with length greater than three have infinite cost.
b. Suffixes have basic cost inversely proportional to the number of segments immediately left of the suffix which are found at the end of the input form.
c. Suffixes have minor (tiebreaking) cost proportional to their length.

An algorithm computing the least-cost suffix according to this scheme somewhat more efficiently than one enumerating all the possibilities is given in (95). Sample runs can be found in (96)–(97).

As I pointed out above, once we have a system for extracting suffixes, we can easily construct a system for computing suffixation error. The suffixation error will be defined as follows: first, use our suffix cost function to find the best suffix given an input pair $\langle u, v \rangle$. Then return this as an error *if* that suffix did not apply in our derivation. This includes the—wholly possible—case in which *no* appropriate suffix can apply in the derivation, and this yields entirely correct surface output—if no rule applies to *cut*, for example, which has past tense *cut*; no suffix applied—not even the null suffix—and this is incorrect, so we send an error signal back (-ø).

Again, we need to determine what the input to this procedure—the pair $\langle u, v \rangle$—should be. The target form, v, should surely be the attested surface past tense. There are two obvious possibilities for u. It could either be the underlying stem form (*ring*, in the case of *ring–rang*), or the predicted internal-change output from our derivation (which might well be *rang* in the case of *ring–rang*). In the case of the current learner, this makes no difference; for concreteness, I will make the latter choice.

3.1.3 Naive Learning

In section 3.1.2, I developed error signals that would tell the learner where it had gone wrong in a particular derivation and inform the learner as to what new rules to add, or how to adjust its existing rules. I noted there that it was impossible to imagine sending the *same* error signal to both the internal change and the suffixation component, since internal changes and suffixes were two different kinds of things, and the learner, after all, needs error information in terms that it can use to generate new hypotheses; if it is trying to put together a collection of apples, information about what sorts of oranges are missing is useless.

I used this argument to justify using some knowledge about the architecture of grammar in the error signal. In particular, I argued that the error for internal changes should not be computed by comparing the learner's final prediction to the input it received, and seeing where the two differ. Instead, it should be computed with respect to the output of the internal change component in the learner's derivation—the input to the suffixation component—and attempting to make the internal component match the input *minus the probable suffix*. This system uses basic information about how grammar works—and thus what its *entire derivation*, not just its final output—should look like, to compute different error signals for each linguistic lesson.

While this seems worthy of little attention by itself, it is plausible that there might be a simpler solution. Yang (2002), examining the prospects for a system learning multiple, interacting parameters, suggests, with some optimism, that we might be able to get away with sending the *same* error signal to *every* parameter. For example, in a system with a ±*pro*-drop parameter and a ±V2 parameter (see Chapter 1, section 1.3), in a system in which the error signal is a measure of whether some stochastically chosen parameter value is good or bad for some input, we might have success with simply checking to see if the learner's final output was correct, and then telling it to update *both* parameters in the same way—favourably, if the final output was correct, unfavourably, if the final output was wrong. Correct parameter values might sometimes get penalized incorrectly in this way, just by virtue of "playing on a losing team," but, suggested Yang, things might nevertheless work themselves out; Yang called this system *Naive Parameter Learning* (NPL).

It is unlikely that anyone would suggest this solution for our two linguistic lessons—the set of internal change rules and the set of suffixes—and, as pointed out above, it is impossible to make the error signals exactly the same, as internal changes as suffixes are somewhat different—but it is possible to imagine making the two error signals somewhat more similar, if we strongly believed that this sort of system was the way to go. In particular, the computations of internal change error and of suffixation error both take place by comparing two forms, so that the error signal is an internal change/suffix telling the learner the difference between what the system did and what it should have done.

To find the suffixation error, the learner compares its derivation with the attested surface form. On the other hand, to find the internal change error, the learner makes a comparison to the form that *should be input to the suffixation component*, given the suffixation error. (See section 3.1.2.) An arguably more naive way to do this, which would rely on less knowledge about grammar, would be to compare our derivation to the attested surface form *in both cases*.

As expected, this is not workable. Indeed, in simulation, the learner fails to inflect any verbs correctly. (See the following section for more details of the training and test sets, and for results to compare.)

The types of errors the learner makes are predictable. Many of the errors can be seen to come from internal change rules like those in (33). (Recall that there is a feature chart in Appendix E.)

(33)
$$\varnothing \rightarrow t/[+\text{cons}, -\text{lat}, -\text{nas}, -\text{son}, -\text{voice}]\_\#$$
$$\varnothing \rightarrow d/[+\text{cont}, +\text{voice}]\_\#$$

The learner arrives at rules like these after seeing several predicted forms which lack either a *-t* or a *-d* suffix. Since the error that is used to update both the internal change rules and the suffixation rules is measured in terms of deviance from the fully inflected form heard by the learner, we can see that, if the predicted form lacks a suffix, then *both* the set of internal changes and the set of suffixes will be modified to suit. Redundant internal change rules like the ones in (33) are thus posited, meaning that the learner is stuck with doubly-suffixed forms like [ɪmprɛstɪd] for *impressed* ([ɪmprɛst]) and [pʊtɪdɪd] for the past tense *put* ([pʊt]). While one can imagine a learner that did not separate internal changes from suffixes, and thus would behave roughly along these lines, that is not the sort of grammar we are attempting to learn. Our grammars strictly order internal changes before suffixation, but the learner does not yet have any way of taking into account this interaction. This is disastrous. I conclude that naive learning is not feasible for all linguistic lessons, and there is no reason to think that the learner might not have an error signal fairly finely tuned to the particular linguistic lesson being learned. I use this as justification for pursing a theory that assumes such complex domain knowledge in a rather more extreme way in discussion of the acquisition of morpholexical marking in section 3.2.

3.1.4 Summary

The learner developed in this section is the basic model we will be developing in the rest of this chapter. It keeps apart two kinds of error signals, in the sense of the introduction to this chapter, because it has two components, which need to be updated in different ways. It uses these signals to update its model of the ambient grammar. It is given in full in (34).

(34) On input $\langle s, t \rangle$:

a. Predict the past tense of *s* using the current set of rules.

b. Based on the attested output, *t*, and the predicted output, compute error signals that will tell the learner how it should change its grammar; in particular, error signals telling the learner what is missing from its rule system (one for internal changes and one for suffixation).

c. Change the grammar using the error signal: use the rule error signals to determine whether new rules should be added, or whether existing rules should be

Trained on	Training	Weak 1 (Unseen)
Mixed 2	64%	100%
Mixed 5 (82-Verb Set)	54%	81%

Table 3.1: Percentage of verbs correctly inflected after ten training epochs.

generalized, or neither.

The reader should be aware that, in this section (and throughout this chapter), I do not intend to present any serious tests of the learner. Instead, I only wish to present tests suggesting that the learner is doing roughly the right thing. The fact that the learner does fairly well (but not perfectly) on weak verbs is only meant to tell us that it has learned something, but that it cannot sort out weak and strong verbs (the problem to be addressed in the next section). Proper tests on novel verbs are not of the nature given here; see Chapter 4.

Its results in simulation are given in Table 3.1. The first column in Table 3.1 names the data set the learner was trained on. Mixed 2 is a collection of 28 English verbs, some weak (including all three -*d* allomorphs) and some strong. Mixed 5 is a larger collection of weak and strong verbs (82 verbs). The second column shows the learner's accuracy inflecting verbs from its own training set after being exposed to all the verbs in that set ten times over.[4] The third column shows the learner's accuracy inflecting a collection of 20 unseen English weak verbs after these same ten exposures.

The learner's performance is relatively good (particularly compared to that of the learner in section 3.1.3), showing that it is, in fact, learning something.

Examining the output reveals that the learner does not make any of the types of errors reported in section 3.1.3, but it still makes mistakes. Trained on Mixed 2, virtually the only type of mistake it makes is to mark many strong past tenses with a gratuitous -*d* suffix (*catch–caughted*, *sing–sanged*). Trained on Mixed 5, the learner also begins to mark weak past tenses with the null suffix (*surrender–surrender*), because Mixed 5 is large enough to give the learner evidence for a much more general environment in which the null suffix applies. The most telling of its errors, however, is its failure to distinguish *hang–hung* from *hang–hanged*, marking both in the

[4] One run through the training data is usually enough to get this learner's grammar to stop changing, so ten times should be more than sufficient.

same way. All these errors point to a single problem: the learner cannot handle morpholexical rules. We will address this issue in the next section.

There is one serious problem with this learner which does not show up in the table of results, however, and which we will not address here. Like many other simulations of the past-tense learning task, this learner posits three separate suffixes ([t], [d], and [ɪd]) for the weak *-d* past tense, although they are in complementary distribution. This is because of the restriction we set out for ourselves in Chapter 1 to the effect that we would deal only with internal changes and suffixes. In order to treat all instances of the weak past tense as deriving from underlying [d], we would need to introduce an extra set of rules with the capacity to change suffixed material.

There are two reasons that this would not be trivial. The first, which seems to me more easily dealt with, is that, although it is fairly obvious how the learner can avoid duplicating suffixes in the internal-change system, as we have discussed here, it is far less obvious how the learner could tell which rules are to be made internal changes and which ones are to follow suffixation. There are some obvious approaches we could take to this problem—I will leave my speculation to Chapter 4. The second problem seems more serious: what would ever make the learner consider the possibility that these three allomorphs are related? This seems to have something to do with phonotactics, but, even if we could get the learner to acquire the requisite phonotactic repair rules, it would need to apply the [d] suffix to a form terminating in a segment inappropriate to the [d] allomorph in order for them to ever apply—and that would never happen using MGL as we have implemented it here.

One solution might be to add to the learner's generalization procedure a system which would generalize [d] and [t] allomorphs into an underspecified suffix, [D], which would need repair by later rules; then, supposing the [ɪd] suffix were present in the grammar, if there were some means of making it *fail* to apply—or at least fail to apply before [D]—then [D] could apply, allowing epenthesis to take place. The [ɪd] suffix would then need to come to be dispreferred, perhaps along the lines of the learner we will propose in section 3.3. Another solution, assuming the necessary phonotactic rules were already known, might be to optionally allow these rules to apply in reverse to augment the suffixation error when they could possibly have generated the attested output: when the [t] allomorph is seen for the first time, the current suffixation-error procedure will currently immediately assume it is new, but if there were a phonotactic devoicing rule which could also have yielded [t], a procedure recognizing [t] as a possible output of this rule could revise the target surface form *v* to contain the necessary input, [d], instead of [t]. And another solution, of course, would be to abandon MGL. We will leave these issues aside.

3.2 Learning Stem-Marking

Recall from the previous chapters the notion of a *morpholexical* pattern. This is a pattern that holds only of a certain arbitrary class of forms. In our terms, it is a rule that will only apply if the lexical entry for the form it is operating on is specially marked with a grammatical feature, which we will call an inclusion feature. We might consider the null-suffix for the English past tense a morpholexical rule, for example, applying only for strong past tenses like *hung*, the past tense of *hang* (as used to mean anything other than a method of execution) and *lay*, the past tense of *lie* (as in *lie down*)—but not for *hanged*, the past tense of *hang* (as used to mean a method of execution), for *lied*, the past tense of *lie* (meaning *prevaricate*), or for *laid*, the past tense of *lay*. Clearly, no phonological generalization can capture these differences, and so, in general, a marking is associated with not only the stored phonological representation, but with the lexical entry.

Although there is some debate in the literature about the mechanism responsible for processing morpholexical patterns (see Chapter 2), it is clear that the learner must at some point *decide* whether a particular pattern is morpholexical or not. If we are going to develop a learning system that operates on high-level objects like *rules* and *lexical entries*—and, indeed, *morpholexicality*—we are going to need some explicit criteria for deciding whether a rule should or should not be morpholexical. (Contrast this with the network model of Joanisse and Seidenberg (1999), which makes exactly this decision implicitly by treating semantic information as just another feature to be weighted more or less strongly by the model's domain-general, low-level learning rules.)

From our point of view, whether a rule is morpholexical or not is a linguistic lesson to be learned. We can think of it as a *selective* problem, in the sense of Chapter 1, and, as such, we can use some theory of selective learning. *Stochastic Grammoid Theory* (SGT), which supposes that learning a linguistic lesson consists of making the *belief strength* of incorrect grammoids go to zero, gives us one such theory straightforwardly.

In section 3.1, I extended to phonological rule systems a very general scheme for learning, of the kind discussed in Chapter 1, section 1.3. Learning a grammar, I pointed out, is a matter of learning a mapping between internal representations and utterances. Under the general view of learning I assume, after the learner has heard an utterance, and deduced the associated internal representations, it then uses its current grammar to map those representations *back* to utterances. It then computes some *error signal* by examining its own output and attempting to determine where it went wrong, and uses this information to adjust its grammar.

Under SGT, this error signal must be used (perhaps among other things) to decide what the change in the learner's beliefs should be. Here, the linguistic lessons to be learned are simple two-way true-or-false questions and so the error signal will be used to determine the change in the learner's belief in the correct answer. In particular, we will need one lesson for each rule determining whether or not that rule is morpholexical; for each lexical entry which we believe might be stem-marked for some morpholexical rule, so that that rule applies to it, we will need a lesson determining whether that stem is indeed marked for that rule; and, for completeness, I introduce here the possibility that a stem may be marked as being an *exception* to some non-morpholexical rule, so that that rule does *not* apply to that stem. All these grammoids will be adjusted stochastically.

The learner we have developed already can easily be extended to do this. The general idea is given in (35), with new features in **bold**. For clarification of the existing behaviour of the learner, see the rest of this chapter and the examples in Appendix D.

(35) On input $\langle s,t \rangle$:

a. Predict the past tense of *s* using the current set of rules.
Because our beliefs are stochastic, we must now choose some current set of:

  · **Rule morpholexicality grammoids, which tell us whether each of the rules is morpholexical.**

  · **Features on *s* determining whether it should be included in a rule's domain (for morpholexical rules) or excluded (for non-morpholexical rules).**

b. Based on the attested output, *t*, and the predicted output, compute error signals that will tell the learner how it should change its grammar:

  · Error signals telling the learner what is missing from its rule system (one for internal changes and one for suffixation).

  · **Error signals for rule morpholexicality grammoids, which tell the learner the direction in which belief in those grammoids should be adjusted.**

  · **Error signals for the rule features on *s*, telling the learner the direction in which belief in the existing grammoids should be adjusted, or whether new ones should be added.**

c. Change the grammar using the error signal:

· Use the rule error signals to determine whether new rules should be added, or whether existing rules should be generalized, or neither.

· **Use the rule morpholexicality error signals to adjust belief in the morpholexicality of each rule.**

· **Use the current rule feature error signals to adjust belief in existing rule features, or to add new rule features to *s*.**

Whatever our criteria for determining whether a rule should or should not be morpholexical, and for whether a stem should be marked in some way for some rule, these criteria will be incorporated in our learner in the form of error signals.

Unfortunately, although morpholexical patterning has been the subject of a good deal of debate, the issue of what exactly the criteria are that learners use to determine whether a pattern is morpholexical has apparently aroused little interest. The only proposal that I am aware of can be found in Yang 2005: learners begin with the assumption that rules not morpholexical, and gradually accumulate lists of exceptions to each. They also (at least initially) keep track of the words to which that rule does apply. Once a rule's list of exceptions starts to take too long to process, on average, compared with the list of words it applies to, learners decide that the rule is morpholexical—that is, they favour the list of words the rule applies to over the list of words it does not. Yang goes on to suggest that the time it takes to find a word in a list of exceptions or inclusions should be equal to the total time it takes to process all the words above it in the list, and that the time it takes to process each word should be inversely proportional to its frequency. Thus lists that are too long (high type frequency) or have too many infrequent words (low token frequency) should be dispreferred.

The proposal I will take up here is inspired, very generally, by this proposal, in that it is supposed to relate the general reliability of a rule to the learner's belief in that rule's morpholexicality.

Taking up a proposal like this, however, means filling in some details. The first detail to be filled in is to answer the question, *When does the learner construe a word as an exception to a rule?* Clearly, it is not correct to assert that a word is an exception to a rule simply because that rule fails to derive the correct output when it applies: other rules might have applied, or the necessary rules might not have applied. It is hard to imagine how we could be certain that a word was an exception to a particular rule. Rather, I will introduce into the learner's error computations a good deal of knowledge about how a "well-behaved" grammar should work, in the form of some heuristics intended to determine *why* a particular rule did or did not apply.

In this section, I will do two things before presenting data about the performance of this new learner, which I will call **ML**. First, I will discuss at length the heuristics that will go into the error signal for the learner's belief in the morpholexicality of a rule, in inclusion marking on some stem for some rule, and in exception marking on some stem for some rule (section 3.2.1). I will then discuss the means by which the learner will use these error signals to update its beliefs (section 3.2.2).

3.2.1 Adjusting the Set of Markings

In order to do anything, ML needs error signals. As discussed in the introduction to this section, in the case of stochastic grammoids, this error signal will be a piece of information about a derivation and its output that tell the learner in which direction to adjust its beliefs.

In section 3.1, I developed error signals to give to the learner for the purpose of updating its set of rules. There, we had two error signals: one telling the learner the kind of internal change it should account for, and one telling it the kind of suffixes it should apply, to repair the derivation it carried out on the most recent input.

Here, we first need to define the error that will be used to adjust the morpholexicality of some rule. As discussed above, this will be calculated based on some general heuristics about what an ideal derivation should look like. These heuristics are given in (36) for those readers uninterested in reading the motivation behind them; these readers may skip ahead to the error signals for stem markings, which are treated in the next major part of this section. For examples of (36), see (105)–(107).[5]

(36) **Belief that a rule is morpholexical will be encouraged (morpholexicality deemed good):**

If a morpholexical rule skips a form (it would have applied but didn't because the stem was not marked for the rule) and this does not lead to an error; for example, if *lick* is chosen without any rule features, and the morpholexical rule ɪ → ʌ was selected, it will not apply, and this is correct, so we encourage this behaviour.

[5] Note that our error signal—our collection of heuristics—is not intended to be used as a reward or penalty to the *current* chosen grammoid. This point will be taken up briefly in section 3.2.2.

·If a non-morpholexical rule applies *after* a previous non-morpholexical rule skipped a form (it would have applied but didn't because the stem was marked as an exception to the rule); for example, supposing there are two suffixes, *-d* and -ø, applying in this order, and only the latter is supposed to apply to *ring*, then, if *ring* is a marked exception to the rule, we should also consider the hypothesis that -ø is morpholexical, and we encourage this when we notice that -ø applied because *-d* was skipped over.

·If a non-morpholexical rule applies and there is a rule error in its component (internal change or suffixation); for example, if the ɪ → æ rule applied to *bring* because it was not morpholexical, it would be encouraged to be morpholexical, because it would lead to internal change error.

Belief that a rule is morpholexical will be discouraged (morpholexicality deemed bad): ·

If a morpholexical rule skips a form and this leads to an error with the same change as the rule under consideration; for example, suppose that the suffix *-d* were deemed morpholexical, and it failed to apply to *fry* as a result; then we would discourage it from being morpholexical.

·If a non-morpholexical rule applies and there is no rule error in its component; for example, if the suffix *-d* applies to *fry* and yields *fried*, the rule's morpholexicality will be (further) discouraged.

This minor section will be devoted to explaining the reasoning behind the error signal in (36).

I will assume that ML keeps track of some basic facts about its derivation, including whether, for each rule, that rule applied, and, if it did not apply, whether it *could* have applied, but failed to because the stem was marked the wrong way for that rule. Being marked the right way for a morpholexical rule means having the inclusion feature that would allow that rule to apply. Being marked the right way for a non-morpholexical rule means *not* having the exception feature that would prevent that rule from applying (see the introduction to this section). I assume that there is no prediction error—that is no change—for any rule that did not apply, and could not have applied. If a rule did not play a role in the derivation, we have no information about it.

An important fact about a derivation is *whether it led to an error*. In particular, recall that in section 3.1, I developed an error signal for the internal change component, which could sometimes be the zero rule ø → ø, in which case no change would be made to the set of internal change rules; I also developed a system for finding the suffixes on past tense forms, but these suffixes were not considered to be suffixation errors in case the same suffix had applied in the course of the derivation. In these cases I will say that the derivation *did not lead to an error* (internal-change or suffixation, respectively). We will use this information in our heuristics.

The first heuristic will be that *morpholexical rules should be skipped over*. The intuition behind a morpholexical rule is that it is a rule that only applies to a small collection of forms. Most of the time, therefore, a morpholexical rule—that is, a rule chosen to be morpholexical on that derivation—should be failing to apply to forms because they are not marked for that rule, and this should cause no problem—that is, there should be no error in that rule's component. If a rule is chosen to be morpholexical, and it fails to apply to a form because the stem was not marked, and there is no error in its component, then belief in morpholexicality for that rule will be deemed *good*.

For example, the rule taking *lie* to *lay* would match the phonological form of *lie*, "prevaricate," and, if it were morpholexical, fail to apply only because *lie*, "prevaricate" was not marked for the rule. It would be correct to see this as encouraging that rule's morpholexicality. On the other hand, if the same rule failed to apply to *lie*, "recline," because *lie*, "recline" was (incorrectly) not marked for the rule, there would be an error, and so we would not encourage the rule's morpholexicality.

We would not always *dis*courage a rule's morpholexicality just because we did not encourage it, of course. In particular, if a morpholexical rule *does* apply to some stem, we will not take this as evidence for that rule not being morpholexical (that would not make sense). In fact, we will not take it as evidence at all: a rule might be made morpholexical and have many forms included, but then prove to be more general than previously suspected—just because it is *possible* to come up with a consistent grammar under the assumption that some rule is morpholexical does not mean that is necessarily the grammar we want.

We will, however, have a second heuristic for certain cases in which the first fails: *morpholexical rules should not be* ***wrongly*** *skipped over*. If a morpholexical rule is skipped over because the stem was not marked, then morpholexicality will be considered *bad* for that rule *if the rule's component has an error with a change the same as the rule's change*. For example, if we had made our -*d* rule morpholexical, it would be frequently skipped over, and an error would result. Furthermore, we would know that error to be due to the failure of the -*d* rule to apply, because the error would have the same change as the rule that should have applied.

The third heuristic will be that *non-morpholexical rules should apply without resort to trickery*. Recall that stems can be marked as exceptions to non-morpholexical rules. Suppose that we believed both -ø and -*d* to be non-morpholexical suffixes. If the only time that -ø was ever able to attach to a stem was when stems were marked with exceptions to -*d*, we would suspect something was wrong. Each time one non-morpholexical rule is skipped over because of an exception feature on the stem, and then another non-morpholexical rule applies, we will take morpholexicality to be *good* for the second rule.

This criterion is really only appropriate for disjunctively ordered rules, not for conjunctively ordered rules. For disjunctive rules, we know, when there was an earlier rule that could have applied but did not, that, had that rule applied, then *any later rules would have been blocked*. This is really what we are interested in—whether the second rule's application was only allowed because the first rule was skipped over—and so, for conjunctively ordered rules, a more sophisticated system is probably in order. It is largely irrelevant here, however, since I expect there to be few cases, if any, in which the criterion will be relevant for the internal change rules in the English past tense system.

The final heuristic will be, we may expect, very approximate, but perhaps a good baseline: *non-morpholexical rules should not give rise to errors*. Whenever there is any error in a non-morpholexical rule's component (internal change or suffixation), morpholexicality will be considered *good* for that rule. Whenever there is no error, morpholexicality will be considered *bad* for that rule.

Note that one kind of exception we will be careful *not* to take as evidence for morpholexicality is when a non-morpholexical rule is skipped over because of an exception *marking* on a stem for that rule. (We will encourage *later* rules to be morpholexical, but not the rule itself.) Although *morpholexical rules should be skipped over*, I believe that making this move would give a wrong prediction about the relation of morpholexicality to the frequency of exceptions.

Clearly, a high frequency of exceptions will be partly responsible for a rule's eventually being made morpholexical under our scheme. However, the relevant frequency will not be simple token frequency of exceptions. If an individual word is an exception to a non-morpholexical rule, under the current scheme, it will encourage that rule's morpholexicality, because the rule will apply with an error. Soon, however, we expect (once we have developed a proper scheme for marking exceptions) that the word will be marked as an exception to that rule. If, as I claim, marking a word as an exception to a rule *prevents* further occurrences of that word from making the rule morpholexical, then the relevant predictor of morpholexical status will be roughly *type* frequency, as in Yang's (2005) proposal, which I take to be sensible in this respect.[6]

[6] Unlike in that proposal, we do not expect that a rule with a few exceptions each having very low token frequency should become morpholexical—an empirical question, of course, but one which it is difficult to imagine answering without substantial time and resources.

On the other hand, if we allowed a marked exclusion to count as evidence for morpholexicality, we would expect any non-morpholexical rule with one frequent, marked exclusion to always eventually be made morpholexical, which I take to be wrong. We will thus do nothing when a non-morpholexical rule is passed over because of an exclusion feature.

Note, however, that if a rule *is* chosen to be morpholexical, then its morpholexicality *will* be encouraged in proportion to the token frequency of exceptions, because there is no way of preventing the first heuristic from making the rule morpholexical if it skips over an exception without issue.

This concludes the justification of our error signal for a rule's morpholexicality lesson.

We now turn to the other important error signals here—the error signals which affect whether individual stems will be marked as inclusions or exceptions to individual rules. I will assume that, for each stem, there is, or can be, for each rule, a lesson determining whether that stem is marked as an inclusion to that rule, and a lesson determining whether that stem is marked as an exception to that rule.[7]

The error signals used by ML to update these lessons are given in (37). Inclusion error will only be set for a rule if the rule was chosen as morpholexical. Exception error will only be set if the rule was chosen as non-morpholexical. (Otherwise, these errors are irrelevant.) As before, the uninterested reader may wish to stop here, and proceed to the next section. Examples can be found in (109)–(111).

(37) **Inclusion good:** Morpholexicality was deemed *bad*.

Inclusion bad: Morpholexicality was deemed *good*.

Exception good: Morpholexicality was deemed *good*.

Exception bad: Morpholexicality was deemed *bad*.

We begin with the error signal to be sent to inclusion features for a rule (only in case that rule was chosen as morpholexical). Recall our heuristics from above: *morpholexical rules should be skipped over* and *morpholexical rules should not be wrongly skipped over*. So long as we are using an error signal for morpholexicality along these lines, we can expect that, if morpholexicality is encouraged, it will be because there was evidence for skipping over that rule for the given stem. Thus we do not want to include the stem in that rule's domain of application (things are fine as they are), and we discourage inclusion features. We can also expect that, if morpholexicality is discouraged, it will be because there was evidence that the given stem was incorrectly excluded from the rule. It should therefore be included. This seems reasonable, although it sounds counterintuitive.[8]

[7] I will assume that these lessons are not "constructed" until the first time the learner attempts to encourage belief in the presence of these markings (that is, when the lessons' beliefs are to be shifted towards the *true* grammoid). I do not believe that there is any real empirical force to this assumption; I make it only to speed up simulation.

[8] It would be interesting to see if this criterion would still be effective—and, indeed, it is effective, as we will

The error signal to be sent to exception features for a rule (again, only in case that rule was chosen as non-morpholexical) is just the opposite. It relies on the heuristic *non-morpholexical rules should not give rise to errors*. Again, we expect that any error signal consistent with this strategy should advise the grammar in favour of making the given non-morpholexical rule into a morpholexical one when it gives rise to an error, and against making the given rule a morpholexical one when it does not give rise to an error. Another way of making a rule avoid the same error—rather than making it skip over mostly everything, as it would if it were morpholexical—is to mark an individual exception to the rule.

All these error signals are imperfect, but, as we will see, they do what we want them to. They rely quite heavily on the learner knowing how morpholexical (and non-morpholexical) rules should behave (a move vaguely justified by the failure of a naive, domain-general error signal in section 3.1.3). In the next section, we will discuss how the signals are interpreted by the learner.

3.2.2 Updating Beliefs

In section 3.2.1, I developed a set of domain-specific heuristics that together made up the error signal for adjusting the belief strength of the grammoids for the new linguistic lessons introduced in this section. These were to be used in a way consistent with the kind of system discussed in Chapter 1 (section 1.3), with the beliefs in each grammoid for each of the linguistic lessons forming a probability distribution, to be gradually shifted, with the help of the error signal, towards a grammar consistent with the data. We did not say what system we would use for updating belief strengths, however.

In Chapter 1, I introduced the *Linear Reward–Penalty* (L_{RP}) system, used in Yang 2002. This system is useful and well-known, and has interesting properties (see Narendra and Thatachar 1989 for some discussion from the perspective of machine learning), but it has features that make it less than ideal in a learning environment like ours. There are other ways of updating beliefs, but in this section, I will propose a quick-fix solution to the problems of L_{RP}. This is a minor section.

see—if the error signal for morpholexicality were computed differently, since our justification here rather depends on the details of the morpholexicality error signal; but if the morpholexicality error signal is representative of what morpholexicality really "means," perhaps the same strategy should work with *any* good morpholexicality error signal. The same is true of the error signal for exception markings. I leave this issue here.

Recall the L_{RP} scheme, repeated here as (38).

(38)

$$
\begin{array}{ll}
\text{Selected grammoid } G_i \text{ derives input:} & \begin{cases} B_{t+1}(G_i) = B_t(G_i) + \gamma \cdot (1 - B_t(G_i)) \\ B_{t+1}(G_j) = B_t(G_j) - \gamma \cdot B_t(G_j) \\ \text{(for alternatives, } G_j \text{, to } G_i\text{)} \end{cases} \\
\text{Otherwise:} & \begin{cases} B_{t+1}(G_i) = B_t(G_i) - \gamma \cdot B_t(G_i) \\ B_{t+1}(G_j) = B_t(G_j) + \gamma \cdot (\frac{1}{N-1} - B_t(G_j)) \\ \text{(for alternatives, } G_j \text{, to } G_i\text{)} \\ N = \text{total number of grammoids} \end{cases}
\end{array}
$$

Using the L_{RP} scheme given the error signals in this section would be somewhat different from the way we used it in Chapters 1 and 2, because the error signals we have developed are somewhat different from the traditional error signals in systems using L_{RP}. Ordinarily, the system takes some *action* at some point in time, and the action taken results in feedback *about that action*. This feedback is used to reward or penalize the system's belief that it should take that same action again. Here, the actions are choices of grammoids—but the feedback we obtain is not about the performance of an action generally; rather, it is a piece of information computed (in a different way depending on the action taken, or, more generally, on various features of the derivation) about what the correct action (grammoid) really is. The fact that the error signal is not a generic piece of information about whatever action was taken, however, changes little, as we could always restate the signal, for each grammoid, in terms of the chosen grammoid. Here I will not bother to do so, but the reader should keep in mind that, here, the error signal is always a piece of information about how to update the grammoid *true*, not about the chosen grammoid (unless that grammoid happens to be *true*).

More important here is that the L_{RP} system causes problems. When the L_{RP} scheme rewards a grammoid, it does so by adding to its belief strength a proportion (γ) of the remaining belief strength. This means that, if a grammoid's belief strength is large (and thus the remaining belief strength is small) then rewards will be small. On the other hand, when the L_{RP} scheme penalizes a grammoid, it does so by subtracting from that grammoid's belief strength a proportion of *its own* belief strength. This means that, if a grammoid's belief strength is large, then penalties will be large. This scheme loads the dice in favour of abandoning the current hypothesis when it sees even a single counterexample.

To make this clear, let us take an example. Suppose that a learner has a belief strength of 0.95 for a particular (correct) grammoid (say, a two-way parameter) and then gets a conflicting error signal. Then the belief strength of the opposing grammoid will receive an increase of $\gamma \times 0.95$—making the new belief strength for the correct grammoid drop to .90 for a relatively small value of γ (near 0.05). On the other hand, if evidence now arises in favour of the correct hypothesis, its increase in belief strength will be meagre by comparison: $\gamma \times .10$—resulting in a new belief strength of only .905 for $\gamma = 0.05$. Moreover, the closer we thought we were to being certain, the larger the negative effect of a counterexample will be. We will call this property *Hasty Retreat*.

Hasty Retreat is no problem if the learner gets consistent information—but it is a serious problem for ML, which updates its belief strengths based on rather imperfect information—an error signal that is really only a guess about what is correct. It often happens, therefore, that ML receives a good deal of conflicting information about a particular linguistic lesson. For example, a particular rule might be encouraged to be morpholexical if it applied to some form and gave rise to an error, but it would then be deemed non-morpholexical if it applied to a different form without giving rise to any error—our hope is that the right information will get to ML most often, but this is hard when the evidence *against* current beliefs counts for more than evidence *for* current beliefs.

In fact, Hasty Retreat is unworkable in an environment in which more than one lesson is to be learned simultaneously, because such an environment is bound to be noisy—that is, quite often lead the learner towards incorrect local maxima—in the absence of perfect cues. Indeed, the problem was also pointed out by Yang (2002). The solution proposed there was to wait before making any changes to the belief strengths, keeping track of the net number of rewards minus penalties that *would* have been assigned, had there been any changes made, and only making changes when the balance was tipped too far towards a reward or penalty.[9] In the case above, setting a minimum of two changes needed in either direction before altering beliefs would prevent the belief strengths from changing at all after seeing only one piece of evidence *against* the current hypothesis, and one *for*: the single penalty to the correct parameter value suggested (but only suggested) by the inconsistent error signal would be counterbalanced by the next piece of information, consistent with the correct hypothesis—it would be as if nothing happened. This would thus have the effect of avoiding changes when the beliefs are close to certainty.

However, while I cannot deny that, with the appropriate tipping point set, this might solve our problem, in my experiments with this system (on older versions of the learner), I found myself unable to discover a tipping point that was of much help. Furthermore, I believe that the added complexity of this system would only be warranted if we wanted very much to hang on to L_{RP}—but one of Yang's arguments in favour of L_{RP} was, at least in my opinion, that it lent itself to nice analysis, allowing predictions to be made about the expected strength of a learner's belief given information about frequencies in the input. As we saw in Chapter 2, the niceness of this kind of analysis disappears in a realistic, multiple-lesson situation, and adding the balance system (Yang calls it "batch") certainly could not help. I see no reason

[9] Yang's version assumed a uniform error signal for all lessons, and thus only one "balance" to keep track of, but it is equally straightforward to adapt the idea to the current system by adding a counter to each lesson.

to extend L_{RP} in this way rather than simply replacing it.[10]

Luckily, there is a much simpler system than L_{RP} which does not have the undesirable Hasty Retreat property: modify belief strengths by a constant. If ML encountered a counterexample to a correct belief held with 95% certainty, it would decrement its belief strength by some constant—call it the *learning rate*—say, 5%, to 90%. If it then got information favouring the correct belief, it would increase its belief strength *by the same amount*—in this case, from 90% back up to 95%. The same constant would be added or subtracted, no matter how strong the learner's belief, up to the ceiling of 100% certainty in some parameter value. Simulation shows that this leads to more reasonable behaviour in the learner: it no longer retreats from well-supported hypotheses on the grounds of a few unexpected error signals.

In practice, some problems remain, however. Although ML does not have as severe a problem with inconsistent information, it still occasionally gets it—and when it does, its beliefs, of course, change slightly, to disfavour correct hypotheses. This gives rules which need not be made morpholexical the opportunity to accumulate inclusion features and thus fare better as morpholexical rules, with the result that rules are either made morpholexical unnecessarily or are extremely indecisive about their status so long as there is some occasional inconsistent information.[11] The solution is to go even *farther* away from Hasty Retreat. Hasty Retreat is a property whereby learners make larger steps toward having similar belief strengths for both alternate hypotheses—that is, towards greater uncertainty, or *high entropy*—than they do towards supporting already-well-supported hypotheses. A constant reward/penalty system fails to have the Hasty Retreat property—but I suggest that we should go farther than this. What is most useful, I suggest, is to endow the belief updater with the *reverse* property, which we may call the *Anti-Entropic* property: beliefs should tend *away* from high entropy. The closer a belief is to being certain, the *smaller* its moves away from certainty should be.

One way of implementing this, assuming constant (rather than linear) adjustments, is to make the learner attempt to remove uncertainty in its spare time—that is, on iterations which do not provide any evidence about a particular parameter value. This is shown in (39). (G_i is some value i for some parameter G and $B_t(G_i)$ is the strength of the learner's belief in that parameter value at time t; γ is the learning rate and ε is the anti-entropic force, both small constants, with $\varepsilon < \gamma$, since otherwise the gradual attraction to certainty would have a greater effect than the real learning. For an example, see (102).)

[10] Yang also argues for L_{RP} on the grounds that its asymptotic behaviour, which discussed in Chapter 1 (section 1.3) has some empirical support in certain other domains of human learning, and that lends itself to interesting analysis of the study of historical change, so we might indeed lose something by dropping it—but I have no intention of taking the proposal I make presently seriously as a long-term solution either. I would be more than open to finding another, more virtuous system, so long as it lacked the Hasty Retreat property.

[11] Turning down the learning rate—which we might expect to resolve the problem by making erroneous changes in favour of morpholexicality more subtle, giving more opportunities for the correct hypothesis to be restored and fewer opportunities for unnecessary inclusion features to be marked—makes matters much worse, as the learner begins to make decisions much too slowly, leading to erratic behaviour.

(39)

$$\text{To reward } G_i: \begin{cases} B_{t+1}(G_i) = B_t(G_i) + \gamma & \text{to a maximum of 1} \\ B_{t+1}(G_j) = B_t(G_j) - \frac{\gamma}{N-1} & \text{for } j \neq i, \text{ to a minimum of 0} \\ N = \text{total number of grammoids} \end{cases}$$

$$\text{To penalize } G_i: \begin{cases} B_{t+1}(G_i) = B_t(G_i) - \gamma & \text{to a minimum of 0} \\ B_{t+1}(G_j) = B_t(G_j) + \frac{\gamma}{N-1} & \text{for } j \neq i, \text{ to a maximum of 1} \end{cases}$$

$$\text{Otherwise}: \begin{cases} \text{Stochastically choose a value for } G \text{ (as if we were choosing a grammoid), twice.} \\ -\text{If both values are the same, do nothing.} \\ -\text{If the values differ, stochastically select a third value for } G, \text{ and reward this} \\ \quad \text{grammoid, using } \varepsilon \text{ rather than } \gamma. \end{cases}$$

The main part of the belief update scheme in (39) is the same as L_{RP}, except that it changes belief strengths by a constant, rather than by a linear function of their current strengths. The extra part of the belief update scheme in (39) is the *know-nothing* case. If we decide that we have no information about the correct value of G after processing some input, we attempt to remove some small amount of our uncertainty about the value of G. We check our certainty by sampling our beliefs twice—if we get back different beliefs, we are evidently uncertain about the correct value of G. If we are uncertain, we adjust our beliefs in favour of one or the other value, selected according to our current beliefs—a selection which will tend, on average, toward the more strongly believed alternative.

If, for example, we decided we had no evidence from the input about the correct value of some binary-valued parameter over which our beliefs were evenly split—exactly fifty-fifty, the maximally uncertain set of beliefs—then we would choose values for that parameter twice according to those beliefs. This would have a relatively good chance (50%) of yielding two different values. If this happened, we would strengthen one or the other of these beliefs—we don't know which, since we are selecting the belief according to our completely uncertain current beliefs—by some small constant ε. Depending on the size of ε, repeating this procedure would more than likely do little to change our beliefs, since both would continue to be selected approximately equally often.

If, on the other hand, we decided that some input provided no evidence about some lesson which we were fairly certain we had learned—say, 95% in favour of grammoid 1, 5% in favour of grammoid 2—then, while we would still choose two grammoids according to current beliefs, comparing them would more than likely not reveal any uncertainty, and we would do nothing. There still would be a *reasonable* chance of choosing two different grammoids, however (4.75%)—in which case, 95% of the time, we would increase our certainty about the correctness of grammoid 1. For sufficiently small ε, we expect that, despite a few changes in the direction of uncertainty, we will mostly make changes confirming the beliefs we already have. According to this scheme, in the absence of evidence, we force our beliefs towards certainty, guided by whatever belief we currently hold most strongly, tending more towards that belief the more strongly we hold it. As expected, simulation reveals that the Anti-Entropic system takes care of all the problematic indecisive beliefs not eliminated by changing to a constant reward scheme and helps the learner avoid local maxima.

3.2.3 Summary

I have been calling the learner developed in this section ML; ML is intended to find which rules are morpholexical, and which stems should be included in the domain of morpholexical rules, or specially excluded from the domain of ordinary rules.

I trained ML on the same collections of mixed weak and strong English verbs as I used in section 3.1.4 (Mixed 2 and Mixed 5). Recall that Mixed 2 was a small collection of verbs which the learner from the previous section was able to learn with about two-thirds accuracy after fifty epochs; the learner got lucky with the set of weak verbs we gave it for testing after being trained on Mixed 2, but, training it on a larger set of verbs (Mixed 5) showed that its generalization performance was actually quite poor.

Expecting the stochastic component to make this learner need somewhat more training than the previous one, I gave ML its training sets in thirty epochs (rather than only ten times as before). For each training set, I tested the learner on that training set, and on the same collection of 21 weak verbs used to test the appropriateness of the deduced grammar in section 3.1.4 (Weak 1). Results of these tests are in Table 3.2.

Trained on	**Training**	**Weak 1 (Unseen)**
Mixed 2	81%	80%
Mixed 5	100%	100%

Table 3.2: Percentage of verbs in the training data correctly inflected after thirty training epochs for $\gamma = 0.05$, $\varepsilon = 0.005$, except for exclusion markings, for which $\gamma = 0.2$, $\varepsilon = 0.01$. Default beliefs were 0.2 for exclusions, 0.05 for inclusions, and 0.1 for morpholexicality.

The learner does appear to need more training than before: after thirty epochs, its performance on Mixed 2 is better than the learner from the previous section, but not perfect, and, after being trained on Mixed 2, its performance on Weak 1 is actually worse than before. Its performance on Mixed 5, however, shows that this is not because we have failed to deliver the improvements we wanted: on the contrary, having this extra data—whether because the data is more diverse or simply because there is more of it—gives the learner perfect performance, on both training and test data. Inspecting the output of the learner shows that it does indeed make rules morpholexical, and precisely the one we expect, and that the learner had trouble with in the previous section: the null suffix.[12]

In the final section of this chapter, I will address the issue of how (and why) the learner might include several environments for the same rule in its grammar.

3.3 Learning Duplicate Rules

Recall from Chapter 2 the idea (from Albright 2002, Albright and Hayes 2003) that certain gradient grammaticality judgments in speakers might be explained by *duplicate rules*: sets of rules containing several, apparently redundant, environments for the same change, whereas in ordinary rule-based phonological grammars, each change $X \rightarrow Y$ is accompanied by a single environment in which it occurs.[13] The example from Chapter 2 is given in (40).

(40) ɪ → ʌ / #*CCC*_
ɪ → ʌ / *CC*_
ɪ → ʌ / *C*_
ɪ → ʌ / _[nasal]
ɪ → ʌ / _[velar]
V → ʌ / _[nasal]

The facts to be accounted for were English speakers' judgments and rates of production of novel past tenses formed on the patterns of strong verbs. For example, presented with a nonce

[12] As before, the internal changes are not relevant here, because they are each attested in so few verbs that their environments will be too restrictive to lead to any errors anyway; experimenting with other data sets showed that the learner was, of course, capable of making internal changes morpholexical, but the errors it produced were often bizarre, and I determined that it was most realistic to bias internal changes towards morpholexicality in the final tests in Chapter 4.

[13] Except as a kind of last resort: say, if a change a → e happens after high vowels and low vowels, but not mid vowels, and we do not believe there is any feature that we can write in our environment that will get us both environments—no possible a → e/_[−mid]—then we will posit two rules instead of one—but it is certainly not usual to posit two rules where a natural simplification to one could be found, as it would be if we posited both a → e/_[+high, −back] and a → e/_[+high], for example.

verb *spling* ([splɪŋ]), English speakers will often give, or accept, *splung* ([splʌŋ]) as a possible past tense. There are various other environments in which speakers will allow this change (ɪ → ʌ) to marking the past tense, and the rate at which speakers will judge each environment good can be explained under several sorts of theories—but we crucially need to assume that speakers in some way keep track of several possible environments for the change for one environment to be "rated" higher than other. One way of doing this is with duplicate rules, as in (40). (See Chapter 2 and Chapter 4, section 4.2, for details.) If we want to try to account for these facts, therefore, we should allow our learner to keep duplicate rules.

The current learner, ML, follows the system in (41).

(41) On input $\langle s,t \rangle$:

a. Predict the past tense of s using the current rules.

b. Based on the attested output, t, and the predicted output, compute error signals that will tell the learner how it should change its grammar.

c. Change the grammar using the error signal, using the MGL strategy.

ML, as should by now be familiar, works by adjusting its grammar after each input, updating it by determining how its current grammar (or some currently strongly believed-in possible grammar) would do at generating the same thing (in this case, the past tense of s, the stem, which we assume the learner can identify). We assume that it has a different sort of error signal, and, of course, a different way of using this error signal to find the new grammar (or to adjust its beliefs towards more likely grammars, given the new information), for each type of linguistic lesson.

To change the set of rules it believes to be in its grammar in response to the error signal, ML either adds a rule, or else combines the environment of the rule it *would* have added (the new rule induced by the error signal) with that of an existing rule to give the existing rule a more general environment for that existing rule; otherwise, it does nothing. Crucially, it always replaces the environment of an existing rule when it can (that is, when the error rule has some particular change, and there is already a rule with that change in the grammar). This precludes the possibility of maintaining two different environments for the same rule, so we must make some change to the current system if we wish to allow duplicate rules. (See section 3.1 for details of the system.)

In fact, there are more pressing reasons to add duplicate rules to the system than the prospect of explaining speakers' judgments. Although we know that the learner, in its current state, performs relatively well on the small corpora we have been using for development so far, it is easy to foresee problems.

Recall our discussion pointing out that the learner is unable to collapse the three allomorphs of the weak *-d* past tense into a single underlying form /d/. Rather, it posits three separate suffixes: [d], [t], and [ɪd]. The voiceless allomorph of the *-d* suffix only occurs after voiceless segments. The MGL scheme developed in section 3.1 should not extend the environment of the [t] suffix to segments that are not voiceless, and this is a good thing, because that would lead to serious errors.

This state of affairs will not last. Up to now, our training data has not included strong verbs like *mean–meant* and *feel–felt*. Once these are added, however, ML will recognize that these forms have a suffix [t] and erroneously combine the environments of this suffix with the environments of the voiceless allomorph of *-d*; after all, it has no way of telling these apart, because its representation of the voiceless allomorph of *-d* is also [t]. This, we predict, will be disastrous, because ML will now believe that [t] can apply in voiceless *and* in voiced contexts generally. This will cause it to make many errors, and, presumably, to make [t] morpholexical, and thus do the wrong thing on easy nonce verbs like *croff*, which are supposed to take the voiceless *-d* allomorph, but will instead fail to be inflected, or will perhaps have some other suffix apply, because of [t]'s across-the-board morpholexicality.

Without attempting to solve the problem of *-d* allomorphy, (thus arguably precluding the need for a solution here), we need to have two *versions* of [t]: one that is morpholexical, and one that is not. This requires us to sometimes add a new rule even when there is already one with the change we would like to incorporate (in this case a suffix).

One simple way to do this might be simply to add every rule to the grammar, as Albright and Hayes (2003) do—that is, after each input, the learner could compute the error, then use the error to generalize existing rules with the same change, but, rather than replacing these rules with their generalizations, simply add them as new rules. If the learner already had a change ɪ → æ/#r_ŋ#, then the error ɪ → æ/#s_ŋ# would cause it to add the generalization of these two rules (ɪ → æ/[+cons, +cont, +cor, −lab, −lat, −nas, −syll]_ŋ#), and, if we followed Albright and Hayes, ɪ → æ/#s_ŋ#, but to leave ɪ → æ/#r_ŋ# in the grammar. This would result in a grammar with many rules (more rules than inputs if we follow Albright and Hayes).

The most obvious problem with this is that the running time of simulation is related to the number of rules; first, because the learner must constantly reevaluate the morpholexicality of most or all of the rules, and, second, because it must then frequently attempt to generalize its error rule against *all* the rules with the same change. This would therefore lead to incredibly slow simulations. Moreover, running such a learner shows that if it can have too many rules that are *not* morpholexical, it can account for too much of the data without resorting to morpholexical marking. This leads it to poor generalization performance.

The solution is to reduce the number of rules, in two ways: first, by not adding so many rules to begin with, and second, by associating with each rule an existence grammoid. If a rule's existence is determined to be *false*, that rule will never be chosen and never apply. This has almost the same effect as removing the rule (we could not remove the rule however, without some criterion for determining when a rule really did not exist, and when it might return on the grounds of some new data) removing some of the low generalization rules that keep the learner from, positing morpholexicality (but not speeding simulation).

This is mostly parallel to our solution to the problem of morpholexical marking in section 3.2, in which we had an error signal telling the learner the direction in which it should adjust existing morpholexicality grammoids. In that case, however, certain lessons had to be constructed (inclusion and exclusion features), and we simply added them the first time the error told us to adjust them.

In this case, we will do our construction similarly to the way we did it in section 3.1: we will construct new rules, along with their existence grammoids, based on the same internal-change and suffixation rule error signals, which tell the learner what kind of change cannot be accounted for in its present system, but we will change the way these signals are interpreted by the learner, so that the learner introduces duplicate rules under certain conditions. We will also add a *second* error signal telling the learner how to adjust current existence grammoids.

The general idea is given in (42), with new features in **bold**. For clarification of the existing behaviour of the learner, see the rest of this chapter and the examples in Appendix D. I call this new learner **DL**.

(42) On input $\langle s,t \rangle$:

a. Predict the past tense of s using a currently favoured set of grammoids:

· **Rule existence grammoids, which tell us what rules to include in the derivation.**

· Rule morpholexicality grammoids, which tell us whether those rules are morpholexical.

· Features on *s* determining whether it should be included in a rule's domain (for morpholexical rules) or excluded (for non-morpholexical rules).

b. Based on the attested output, *t*, and the predicted output, compute error signals that will tell the learner how it should change its grammar:

· Error signals telling the learner what is missing from its rule system (one for internal changes and one for suffixation).

· **Error signals for existence grammoids, which tell the learner the direction in which belief in those grammoids should be adjusted.**

· Error signals for rule morpholexicality grammoids, which tell the learner the direction in which belief in those grammoids should be adjusted.

· Error signals for the rule features on *s*, telling the learner the direction in which belief in the existing grammoids should be adjusted, or whether new ones should be added.

c. Change the grammar using the error signal:

· **Use the rule error signals to determine whether duplicate rules should be added, or whether a new rule should be added—or neither—then add the necessary rules, giving each an existence grammoid.**

· **Use the existence error signals to adjust belief in the existence of each rule.**

· Use the rule morpholexicality error signals to adjust belief in the morpholexicality of each rule.

· Use the current rule feature error signals to adjust belief in existing rule features, or to add new rule features to *s*.

The rest of this chapter will be dedicated to explaining this new behaviour in more detail. Readers with no interest in the details of these error signals and how they are used may wish to turn to the summary in section 3.3.3 at this point, or, if they still lack a clear picture of what exactly these additions mean for the learner, to the sample run in (87)–(90). Readers with a passing interest in the details may wish to read the major sections of the rest of this chapter, and skip the minor sections.

3.3.1 Limiting the Set of Rules

Recall ML's system for adding rules to its grammar, given in (43).

(43) Given an error signal $C/$Environment (which might be either an internal-change or a suffixation error) . . .

a. If there is already a rule in the grammar with C, use MGL to combine its environment with Environment. (See section 3.1.1 and example (99) for an explanation of MGL.)

b. Otherwise, add $C/$Environment as a new rule.

This system takes a rule error signal (which comes in the form of a rule) and attempts to generalize any existing rule with the same change (in the appropriate component—internal change or suffixation) to include the given environment. As long as there is a rule with a matching change, no further rules with that same change will be added; thus there can be a maximum of one rule with a given change. (For further clarification of this system, see section 3.1.)

In this section, I will modify this system to allow for the learner to have more than one rule with a particular change, while being judicious with these duplicate rules in order to avoid having too many.

To summarize briefly for those readers uninterested in details, I will propose a system like (44). (The reader should recall that I propose to include with each rule a grammoid determining whether that rule exists; the learner stochastically chooses a value for this grammoid for each rule on each derivation, and, if the chosen value is *false* for some rule, then the learner will not use that rule in its derivation. See section (3.3.2) for details.)

(44) Given an error signal Change/Environment (which might be either an internal-change or a suffixation error). . .

a. Find all the rules with change Change, and choose some subset of these rules stochastically using their existence grammoids.

b. Get the set containing the error rule, plus the generalizations obtained by combining Environment with each of the chosen rules. If any of the generalizations in this set is already in the grammar, augment belief in its existence.

c. Add the most general rule from this set to the grammar.

Readers seeking motivation for or further clarification of (44) may wish to read the rest of this section.

A learner must carry out some generalization in order to be *learning* in any meaningful sense. In our learner, when learning what rules should be in the grammar, this means producing more general versions of rule environments, by combining the error rule's environment with existing environments. This, in turn, means choosing some existing rule(s) to be generalized. Before, of course, choosing which rule would be generalized was easy: there was only one other rule with the same change (at most). For DL, however, which has duplicate rules, we have choices.

For one thing, DL might come up with a set of generalized environments, or it might come up with a single generalized environment. In the interest of minimizing the number of rules in the grammar, we should take the latter choice.

We must then determine what this environment should be (which will tell us how to find it). Four obvious choices that come to mind are the widest possible generalization, the narrowest possible generalization, the result of combining the error rule with the most general existing rule, and the result of combining the error rule with the least general existing rule.

For example, suppose that DL had the error $C/$rɪk_ to integrate, where C is some change. Further suppose that it already had the rules $C/[-\mathrm{ant}, -\mathrm{cont}]$_, $C/[+\mathrm{cont}, +\mathrm{syll}, +\mathrm{tns}, +\mathrm{vc}]$_, $C/$plɪđ_, $C/$suk_, $C/$krɑ_, and $C/$pli_ in its grammar (never mind how exactly the existing generalizations got there—that would depend on which system we pick here). I put aside for now the possibility of combining with some subset of the existing rules, or of taking into consideration the existence beliefs for the existing rules, though I will do this below.

Making the narrowest possible generalization would result in the learner generalizing to the new rule $C/[-\mathrm{cons}, +\mathrm{cont}, +\mathrm{high}, -\mathrm{low}, -\mathrm{nas}, +\mathrm{son}, +\mathrm{syll}, +\mathrm{vc}]$k_ (following high vowels), the result of combining with $C/$suk_; making a generalization with the least general existing rule would mean combining with $C/$plɪđ_ (giving $C/[-\mathrm{ant}, +\mathrm{cons}, -\mathrm{cont}, -\mathrm{lab}, -\mathrm{lat}, -\mathrm{nas}, -\mathrm{son}, -\mathrm{syll}]$_) (following non-labial oral stops), since rules with more segments in their environments are less general. In both cases, we would be combining our rule with one of the most specific rules—the ones obtained directly from the error—rather than with a rule that was the result of some previous generalization. This would tend to be the case with either of these strategies. This will lead to narrow environments being added to the near-exclusion of wide environments. This will keep DL from learning very much, and it will lead to a large number of rules. I thus reject these strategies.

We have two ways of making wide generalizations: either to combine with the widest existing rule, or to search for the most general rule that would result from a combination with any of the rules. The former would certainly seem to have a faster implementation (pick one rule and combine with it, rather than combining with a large number of rules, then seeing which is the most general), and the latter could often be approximated by the former. In this case, however, they are different: the most general existing rule is $C/[-\text{ant}, -\text{cont}]\_$, and combining with it would leave it unchanged; the most general possible rule, on the other hand, we would get by combining with $C/[+\text{cont}, +\text{syll}, +\text{tns}, +\text{vc}]\_$, to yield $C/\_$ (everywhere). In order to get the most general rule possible given the data, as fast as possible, and thus reduce the total number of rules generated (once we have the most general rule possible, no new rules will be added), I will opt for the second possibility, and find alternate ways to make DL faster.

Combining the error rule with *all* of the rules has another benefit: it allows us to prevent commonly attested rules from falling by the wayside, by *rewarding belief in all of the resulting generalizations if they are already in the grammar*. In the example given above, although we would only add $C/\_$, we would boost the learner's belief in $C/[-\text{ant}, -\text{cont}]\_$, and in any of the other generalizations that were found to already be in the grammar.

Given this extra computation, however, it is worth attempting to further minimize the simulation complexity of the learner. I will suppose that, rather than attempting to combine each error signal with *all* of the existing rules, DL *chooses* the rules it will combine the error signal with, according to its current beliefs, just as if it were going to make a prediction.

For example, suppose that DL had posited the suffix $[\text{d}]/[\text{sɑb}]+\_$ with some initial belief strength for its existence—we will use zero. The error $[\text{d}]/[\text{muv}]+\_$ would not combine with this rule if observed, since there would be zero probability of selecting the original rule, and thus zero probability that DL would attempt to combine the error with it. Supposing that DL saw the error $[\text{d}]/[\text{sɑb}]+\_$ again, however, it would increase its belief in that rule, since that rule would be one (the only) possible generalization of the error rule with the selected set of existing rules (the empty set, given the initial belief strength of zero).

At some point, therefore, observing the error $[\text{d}]/[\text{muv}]+\_$ again would lead DL to posit the more general rule $[\text{d}]/[+\text{voice}, +\text{labial}]+\_$. This rule, in turn, would eventually come to have significant belief strength if more similar errors were generalized with similar rules—say, $[\text{d}]/[\text{muv}]+\_$ with $[\text{d}]/[\text{sɑb}]+\_$ again—for the same reason as $[\text{d}]/[\text{sɑb}]+\_$. After one of these rules had become sufficiently strongly believed to exist, there would be no more specific rules added, since there would always be something to generalize an error with. In this way I hope to reduce the number of rules in the grammar and reward certain good partial generalizations.

This would, ideally, alongside our policy of rewarding intermediate generalizations, have the side effect of encouraging a stable, core set of well-established rules, because the best rules would rise to the top and be combined with again and again. In practice, this does not seem to be what happens with DL. Many of the rules in the grammar are able to maintain a middling belief strength, and thus the set of rules included in a derivation is typically large, and fluctuates greatly. I currently have no solution to this problem.

In the final section before I test DL, I discuss the error signal that will be sent to the existence lessons.

3.3.2 Adjusting the Set of Rules

Recall that the new system for adding rules (section 3.3.1) gives each rule an existence grammoid, and relies on these existence grammoids to add new rules.

In section 3.2.1, I presented error signals which would tell ML the direction in which it should adjust its beliefs in the morpholexicality of some rule, and in markings for a stem for some particular rule (inclusion or exclusion markings). These rules relied heavily on domain knowledge—we used our linguist's intuition, along with some inspiration from Yang 2005, to develop a signal that we thought would be make the right rules morpholexical, and include or exclude the right stems from rules.

In this section, I will present similar heuristics for the new existence lessons. For readers not interested in the details or the motivation, the criteria are presented in (45).

(45) **Existence good:**

The rule applied, and there was no error in its component.

Existence bad:

The rule applied, and there was an error in its component; the rule was skipped over (because it was morpholexical or the stem was marked with an exception feature) but there was no error, and a rule with the same change applied.

These criteria are used in much the same way as the rules developed in 3.2.1: DL determines whether there was a rule error (an internal change or suffixation error, of the sort we used to update the set of rules in section 3.3.1; see section 3.1). DL then uses the fact that there was or was not an error, along with some data on whether the rule applied, and why it applied or did not apply, as suggestive of whether the rule is worth including in its grammar. The resulting error signal is used to update DL's belief in the given rule's existence as described in section 3.2.1 and section 3.2.2. For examples, see (113)–(114).

The rationale behind these error signals is quite simple. The basic criterion is that the rule not give rise to any error when it applies. We thus consider the existence of a rule *good* when the rule applies with no error, and *bad* when the rule applies, but there is an error in its component. This, like some of the criteria for adjusting stem-marking beliefs presented in section 3.2.1, will fail badly in case there are interacting rules, or, more generally, multiple rules applying, in the conjunctive component—since *all* the rules that applied will get the rap for the problems caused by one rule—but that does not really arise in the English past tense, and so we ignore the problem here.

We augment this system by attempting to eliminate useless rules. Rules are useless if they can be eliminated from the grammar, and we know they can be eliminated from the grammar if they are skipped over, and then a later rule applies and carries out the same change. Again, as in the parallel criterion found in section 3.2.1, this is probably more appropriate to suffixes applying disjunctively, but developing a full system to handle conjunctive ordering and its various possible complications is well outside the scope of this paper—I leave the issue.

I now discuss whether this learner works.

3.3.3 Summary

I have been calling the learner developed in this section DL. Recall from the introduction to this chapter that we expected this learner to do better than ML where the training data includes verbs, like *mean–meant*, that take a [t] suffix despite a voiced final stem consonant.

In order to test DL, I added several verbs of the *mean–meant* class to Mixed 5 (Mixed 6). I then ran both ML and DL on this training data, and compared their performance on unseen verbs (Weak 1).

Trained On Mixed 6	Training	Weak 1 (Unseen)
ML	95%	60%
DL	100%	95%

Table 3.3: Percentage of verbs in the training data correctly inflected after thirty epochs, with parameters as in section 3.2.3; the learning rate for rules' existence is set to 0.2 for internal changes and 0.1 for suffixes, and the anti-entropy force to 0.001.

The presence of the new, inconsistent [t] suffix causes ML to take slightly longer than before to learn the training data (its performance in the last section was perfect after thirty epochs), but it still does well because of the possibility of morpholexical marking. On the same weak test data as before, however, we see a severe decline in performance. Examining the output shows that this is indeed because of the learner's overgeneral [t] rule being made morpholexical.

DL does much better, but, despite our best efforts, deduces many, many rules, some of which are non-morpholexical -ø rules. (It is also quite slow.) These rules are not made morpholexical because there are so many *other* rules in the grammar which together handle most cases.

Note that this learner still shows less than perfect performance on Weak 1, unlike ML trained on Mixed 5, but inspecting its output shows that this is *not* because of the presence of these erroneously non-morpholexical -ø rules. Rather, it is because the introduction of *mean–meant* allows it to generalize that verb's internal change with the change from *keep–kept*; this happens to be consistent with Mixed 6, and, rather than mark these verbs for inclusion in this rule, the learner makes this rule non-morpholexical. I expect this problem to disappear with a larger training corpus, with which this general rule would not be consistent. Nevertheless, DL fixes a major problem with ML which arises because our learner is unable to collapse the three allomorphs of the weak *-d* suffix. We will wait until the next chapter to test whether this learner can simulate gradient judgments.

This concludes our discussion of the major features a learner for the English past tense must have under the current assumptions. In the next chapter we will test the learner against previous systems, and against adult and child data.

Chapter 4

Results and Discussion

In the previous chapter, we developed an automatic learner for simple morphophonological alternations, developed to handle a standard benchmark in the modelling of morphophonological acquisition: the English past tense. While many other learning systems for morphophonology have been proposed, this one distinguishes itself in constructing grammars closely resembling those posited by linguists. In particular, the system learns grammars that:

(46) Treat suffixation and internal changes as separate, sequentially ordered processes characterised by rules, presenting the challenge of learning two interacting systems. Our learner deals with this challenge using simple domain knowledge about the kinds of interactions that are expected.

(47) Handle idiosyncrasies as properties of lexical items, not just of phonological strings; in particular, its grammars handle idiosyncrasies by marking lexical items specially as either being excluded or included in the domain of a rule, along the lines of Chomsky and Halle 1968, Halle and Marantz 1993. It learns these stem-marking features using a stochastic procedure inspired by Yang 2002, Yang 2005.

In this chapter, we will evaluate the learner critically, first by comparison with other models, then by comparison with real adult speaker behaviour, and then with children's errors.

4.1 The Past Tense Task: Comparison to Other Models

Many other automated morphophonological learners have been trained and tested on the English past tense. In this section we will survey the main models and compare ours against them.

The basic criterion for these learners has generally been *performance on the training data*: after exposure to a set of verbs (all the other models use present–past pairs, like our model), if we ask the learner what the past tense of one of these verbs is, does it know? One way of measuring a learner's performance is simply accuracy: what fraction of the verbs the learner was asked about did it get right? Accuracy is usually quite high, but it is not always at ceiling. Another question to ask about a learner's performance is how long it took—how many times did it need to see the training data to reach its highest level of accuracy?

Perhaps the most basic question we can ask, however, is whether the responses the learner gives when probed are even meaningful at all. Sometimes—generally when modelers attempt the ambitious goal of working out low-level representations—the learner's output representations are not very good, and so its answers must be "rounded" to the nearest realistic output. Our learner passes this first, most basic test: it is not so ambitious as to need any optimism about its answers.

We will compare our learner's performance to that of the models developed by Rumelhart and McClelland (1986), MacWhinney and Leinbach (1991), Ling and Marinov (1993), Plunkett and Juola (1999), and Joanisse and Seidenberg (1999). Their performance on their respective training data sets, along with the results for the current learner trained on a new, larger data set, is presented in Table 4.1.[1]

The present learner has three different incarnations: first, one with duplicate rules, as suggested in section 3.3. I have called this learner DL. Second, a learner without duplicate rules, as developed in section 3.2, which I have called ML. Finally, a learner (without duplicate rules) presented with a simplified problem to solve—all three allomorphs of weak *-d* are given in the

[1] For ML and EL, the learning rate that allowed for these results was 0.05, with an anti-entropic force of 0.005, except for exclusions, which were given a learning rate of 0.2 and an anti-entropic force of 0.01. An initial bias in favour of internal-change rules being morpholexical was also needed (initial belief 0.9, versus 0.1 for suffixes) to avoid obviously unattested errors like *repeat–rapoted*; this seems quite reasonable to me, by the intuition that truly general phonological changes would presumably normally not be in the internal-change system, but rather apply after affixation. The initial beliefs for markings were set to one times the learning rate, and, for existence, to zero. All these values were determined using the smaller data set. After beginning the full test on DL, it became clear that the anti-entropic force needed to be decreased, except for exclusion markings; the results reported here for that learner are for an anti-entropic force of 5×10^{-5}.

Model	**Unambiguous Output**	**Training Corpus**	**Reported Training Accuracy**	**Length of Training**
Rumelhart and McClelland (1986)	No	506 verbs, 98 strong (presentation order related to freq.)	Not presented (graph shows less than 100%)	79,800 trials
MacWhinney and Leinbach (1991)	Yes	1404 verbs, 142 strong	99.2%	24,000 epochs (33.6 million trials)
Ling and Marinov (1993)	Yes	same as MacWhinney and Leinbach (presentation order related to freq.)	99.2%	80,200 trials
Plunkett and Juola (1999)	No	946 verbs, 122 strong (presentation order related to log freq.)	99.2%	1250 epochs (.67 million trials)
Joanisse and Seidenberg (1999)	No	600 verbs, 64 strong (prop'l to log freq.)	99.3%	2.6 million trials
ML (Section 3.2)	Yes	2173 verbs, 149 strong, token frequency exactly proportional to Kučera and Francis 1967 frequency, including some present-tense homophones (*hang–hung* and *hang–hanged*, *lie–lay* and *lie–lied*)	99.1%	55 epochs (1.6 million trials)
DL (Section 3.3)	Yes	Same	95.3%	One epoch (28,946 trials)
EL (Section 3.2, easier problem)	Yes	Same, without allomorphy	99.9%	55 epochs (1.6 million trials

Table 4.1: The results of previous learners and of three versions of our learner.

input as [d], rather than [d], [t], and [ɪd]: as discussed in Chapter 3, the learner is not prepared to collapse these three allomorphs, but supposing we could add a good system for doing this, we would like to know how it might perform. I call this learner EL.

Even without this simplification, however, the learner is still reasonably good—ML performs only slightly worse than previous learners—and EL outperforms them all (almost at ceiling). Given the miniscule variation in performance among the previous learners, it seems reasonable to say that the performance of DL is substantially lower, but there is a reason for this—it hangs on to a lot of rules. As a result, it is very slow, and I was only able to complete one epoch of training data. Having seen almost four times fewer examples than the very fastest of the previous learners (Rumelhart and McClelland 1986 and Ling and Marinov 1993), its performance is quite respectable.

Let us now compare our learner to itself. We expect the performance to be better for EL, as it indeed is. Note that, while this simplification might be workable on the grounds that the learner can solve the phonotactic problems with the *-d* suffix independently, for us to assume that the learner will have completely mastered this problem early on is counter to the facts reported in Berko 1958. This study (the classic study that introduced the *wug* test) showed that correct knowledge of the [ɪd] allomorph is not in general acquired until early school age, much later than the other two allomorphs (the same is true for the parallel [ɪz] allomorph of the plural morpheme). The current learner does not show this pattern, but for EL this should go without saying: if we assume that the learner has mastered the allomorphy problem before past tense learning begins, as we do when we present the learner with forms like [wetd] for *waited* ([wetɪd]), we are guaranteed never to see this pattern.

Recall that in Chapter 3, we also suggested (section 3.3) that, because of the learner's inability to collapse the three allomorphs of weak *-d*, it would overapply the [t] allomorph and need to make that suffix morpholexical. This was expected because there are a few verbs—like *mean–meant*—that actually do seem to take a *-t* suffix, not the weak *-d* (the weak allomorph in the case of *mean*, [min], would be [d]). The learner, we supposed, would be unable to distinguish this *-t* suffix from the unvoiced allomorph of *-d*, and would thus generalize the two, predicting strange (and, as far as I know, unattested) errors like *jog–jogt*. Inspecting the output of ML shows that this is indeed the case. The learner quickly develops an overgeneral *-t* suffix and makes it morpholexical—and, before it has confirmed the rule's morpholexical status, makes the expected voicing-mismatch errors. After the rule has been made morpholexical, it does not reach ceiling until it has stem-marked every word that takes the voiceless allomorph of *-d*. EL

Model	**Training Corpus**	**Length of Training**	**Weak 2**
ML	2173 verbs, 149 strong, token frequency exactly proportional to Kučera and Francis 1967 frequency, including some doublets (both *dived* and *dove*) and present-tense homophones (*hang–hung* and *hang–hanged*, *lie–lay* and *lie–lied*)	One epoch (28,946 trials) and thereafter	54.5%
DL	Same	One epoch (28,946 trials)	100%
EL	Same, without allomorphy	One epoch (28,946 trials) and thereafter	100%

Table 4.2: The ability of three versions of our learner to get English weak *-d* past tenses right.

does not suffer this problem, and neither does DL. The performance of the various learners on a set of novel weak verbs (Weak 2) is presented in Table 4.2.

Although I will take a somewhat radical approach to analysing the learner's performance on novel verbs in section 4.2 (where I will suggest that morpholexicality does not predict productivity), it is clear that, under the standard view of productivity, ML is doomed—on new weak verbs taking the voiceless allomorph of *-d*, the learner does not have any suffix at all to apply (not even the null suffix, which it also marks as morpholexical). Note that, in the case of [t], unlike in the case of [ɪd], there is no reason to think that the learner does not acquire the phonotactic rules that would give the voiceless *-d* allomorph early on. Although DL seems able to deal with the problem even when presented with the three *-d* allomorphs in their surface form—inspecting the output shows that it makes morpholexical only the overgeneralized versions of the [t] suffix—it nevertheless seems likely that learners are aware of the phonotactics of English voicing early on. I proceed to discuss the learners' performance generalizing to novel forms.

4.2 Generalization: Comparison to Adults

Some other learners have been tested on their ability to generalize. This is usually tested by presenting the learner with unseen weak (*-d*) verbs. We used this as a rough test of whether we were doing the right thing in Chapter 3, and we presented the learner's performance on unseen weak verbs in section 4.1—but what does this really test? When English speakers are presented with *spling*, they often inflect it as *splung*, not *splinged* (Bybee and Moder 1983, Albright and Hayes 2003)—and if the learner always passes over morpholexical rules for new forms, as it does when it is inflecting the novel weak verbs we have been giving it, it will never do this.

Recall from Chapter 2 our suggestion that a grammar with several versions of the same rule might be best suited to these kinds of facts: the pattern speakers tap to get *spling–splung*, for example, can apparently be characterized by a set of rules like that in (17), repeated here as (48), under the assumption that speakers prefer forms that can be derived by more rules.

(48) ɪ → ʌ / #*CCC*_
ɪ → ʌ / *CC*_
ɪ → ʌ / *C*_
ɪ → ʌ / _[nasal]
ɪ → ʌ / _[velar]
V → ʌ / _[nasal]

Bybee and Moder (1983) suggest that speakers prefer forms with large onsets and final velar nasals when asked whether words could fit this pattern, though they accept smaller onsets and/or non-nasal velars/non-velar nasals; they also prefer the vowel ɪ, though they will accept other vowels. If speakers keep track of multiple environments for the same pattern and gauge the goodness of a pattern by asking, *How many of the possible derivations for the given stem will get me this form?*, it seems clear that speakers would have these preferences given the rules in (48). From an underlying form that had a velar nasal coda like *spling*, for example, speakers would determine that they could obtain *splung* by using the change-ɪ-before-velars rule, the change-ɪ-before-nasals rule, or the change-any-vowel-before-nasals rule—three of the possible past tense derivations for *spling*. On the other hand, since *splig* has only a velar coda, and not a velar nasal coda, only one of its possible past tense derivations would yield *splug*. It seems like our learner should be able to do this kind of tabulation, and it seems like DL, which keeps track of multiple environments for the same rule, should be better.

The only problem to be resolved is that of morpholexical marking. The grammars deduced by our learner contain rules that do not apply to forms unless they are specially marked—but we must account for the fact that the patterns that speakers will extend to new forms are not limited to those that can be stated without exception—*grow* has past tense *grew*, for example, and *blow* has *blew*, but this internal-change rule does not apply to *glow* (past tense *glowed*). Nevertheless, *sprew* does seem to me to be at least a plausible past tense of *sprow* (though I can find no documentation of speakers' behaviour in wug tests on this particular pattern). Under the current assumptions, the rule taking *blow* to *blew* and *grow* to *grew* does not apply to any but a lexically specified set of forms, and thus never applies to new forms. This forces us to assume that the clear presence or absence of morpholexical marking—that is, features including a stem in a morpholexical rule's domain of application—is a property of existing lexical items only; new stems can be assigned morpholexical markings freely, following some separate procedure.[2] Let us suppose that this is correct, and also that our assumption that stems can be marked as being exceptions to individual rules is correct. Then we can imagine that a speaker might respond to a proposed past tense form for a new stem by going through all the possible morpholexical and exception markings, and determining how many of the possibilities would yield that past tense form. If there were more ways to derive the form (that is, if more different rules could derive it—equivalent to saying that more combinations of exclusion and inclusion features could derive it), then the speaker would have more confidence in that form.[3]

This, then, is precisely what we will do: after learning is complete, we will calculate the "strength" of a proposed past tense form for a nonce stem by the simple counting procedure in (49), which we will call *Nonce-Stem Feature Counting* (NSFC). (NSFC applies, of course, to

[2] Basically the proposal of Schütze 2005.

[3] I can easily imagine other assumptions. For example, we might use the idea that speakers have access to this complex evaluation procedure for new forms—which I call Nonce-Stem Feature Counting (NSFC) below—to deny the necessity of a special class of morpholexical rules. Rules that we might think are morpholexical—like the *grow–grew* rule—are really only rules with a very small set of restricted environments, which cannot in general outcompete the number of possible environments for the *-d* rule. In cases of uncertainty, learners could acquire exception markings to force the application of certain rules. If this could be shown to be feasible—perhaps by simulation along the lines of the current work—we could then ascribe to NSFC, rather than morpholexical rules, the work of ruling out the application of restricted patterns to new forms—but this is outside the scope of this paper.

a particular grammar—that is, the result of the stochastic choice we do at each input.)

(49) On input p (the phonological form of some stem):

1. Begin by positing no markings (inclusions or exclusions).
2. The input to the current rule assuming some set of markings m will always be called i_m; let $i_\emptyset$, the input to the first rule assuming no markings, be p.
3. For each rule r in the grammar, in order:
   (a) For each previously posited set of markings m:
      i. Calculate the form that would result from applying r to i_m assuming that p is marked with the inclusion and exclusion features denoted by m, and save this as o_m (if r does not apply this is the same as i_m).
      ii. If adding the appropriate marking for r (inclusion or exclusion, depending on whether r is morpholexical) to m would change whether r would apply, add that marking to m and call the resulting set of markings m'.
      iii. Calculate the form that would result from applying r to i_m assuming that p is marked with the inclusion and exclusion features denoted by m', and save this as $o_{m'}$ (if r does not apply this is the same as $i_{m'}$).
      iv. If we are in the suffixation component of the grammar, and r successfully attached to either i_m or $i_{m'}$ (it could not have attached to both!), then mark m or m' as being finished, so that on the next iteration no more rules will be considered, and no more markings added, to that collection of markings.
      v. Let i_m be o_m and $i_{m'}$ be $o_{m'}$.
4. At the end of the derivation, some of the sets of markings will be invalid, since they caused no suffix to apply. Remove these from consideration.
5. For each remaining set of markings m, o_m should be equal to the final output assuming that set of markings. Some of the o_m's will be the same. For each possible output v, calculate the fraction of the m's such that $o_m = v$. This is the *strength* of v.

This simple algorithm counts the number of possible lexical entries—collections of rule inclusion and exclusion features that give rise to valid derivations—that would yield a given form, normalized to the total number of possible lexical entries.

For example, suppose the grammar contained one morpholexical internal change rule (call it rule 1) and two, non-morpholexical, suffixes (call them rules 2 and 3). Suppose that rule 1 would apply to [pli] to yield [plɪ] and that rule 2, but not 3 (both of which add [d]) would apply to [pli], and both would apply to [plɪ].

If /pli/ was the proposed stem, we would begin by proposing no markings, so that $o_{\emptyset}$ would become [pli] (recall that rule 1 is morpholexical). We would then add an inclusion feature for rule 1, [+1], and let $o_{\{+1\}}$ be [plɪ]. We would then proceed to the next rule, rule 2. It would apply given ø, making $o_{\emptyset}$ [plid]—and making ø finished—but the extension of ø to contain an exclusion feature for rule 2, [−2], would cause rule 2 to fail to apply, so that $o_{\{-2\}}$ would still be [pli].

Similarly for $\{+1\}$: $o_{\{+1\}}$ would become [plɪd] (and +1 would be finished) and $o_{\{+1,-2\}}$ would be [plɪ].

Finally, we would do the same for rule 3, considering $\{-2\}$ and $\{+1,-2\}$ (since ø and $\{+1\}$ would be finished), along with their extensions $\{-2,-3\}$ and $\{+1,-2,-3\}$. The first two would yield [pli] and [plɪd] for $o_{\{-2\}}$ and $o_{\{+1,-2\}}$ respectively, but only the second would be finished at the end of the derivation, so that $\{-2\}$ would be excluded from consideration. The second two sets of markings (yielding [pli] and [plɪ]) would be unfinished at the end of the derivation, and would thus also be excluded from consideration.

We would thus consider the final outputs of ø, $\{+1\}$, and $\{+1,-2\}$, namely, [plid], [plɪd], and [plɪd]; [plid] would have a strength of $\frac{1}{3}$ and [plɪd] $\frac{2}{3}$. In this way, we can get a kind of gradient judgment from the learner, similar to the judgments provided by the humans and computer models in the study by Albright and Hayes (2003).

To test the theory that our duplicate-rule approach is a good model of human nonce-form judgments, I took the strengths reported by DL in this way on the 90 nonce past tense forms from the study by Albright and Hayes, a collection of pseudo-past forms derived by applying English past-tense processes to phonologically well-formed monosyllabic nonce stems, each stem having one weak pseudo-past and at least one formed by another process. I then compared these ratings with the human ratings reported on these verbs by Albright and Hayes, by taking a (Pearson) correlation. Following Albright and Hayes, I took this correlation separately for the pseudo-weak and pseudo-strong forms, since, as those authors point out, because of the preference for weak past tenses among English speakers, if we simply assigned perfect scores

Model	Weak	Strong
DL	0.45, $p < 0.01$	0.39, $p < 0.01$
ML	0.56, $p < 0.001$	0.53, $p < 0.001$
EL	0.42, $p < 0.01$	0.60, $p < 0.001$

Table 4.3: Comparison of three versions of our learner using NSFC to give nonce-form judgments with human judgments, measured by Pearson's r.

to all the pseudo-weak forms, the correlation taken over all the nonce forms would still be quite strong (0.70). Furthermore, to see whether the performance was attributable to the availability of duplicate rules or to NSFC, I also took the strengths reported by the two other versions of the learner tested in the previous section and did the same test. The results are shown in Table 4.3.

The results show that the judgments given by all three versions of the learner using NSFC are correlated with the human judgments, and that these correlations are statistically significant. Suprisingly, DL does *worse* on strong verb predictions than the other two learners. Judging from the fact that there are many pseudo-weak forms that DL gives a rating of zero that the neither of the other learners rate at zero, it seems clear that this is related to this learner's ability to deny that rules exist; if it denies that some of the more general internal changes exist (and inspecting its output shows it indeed does), then it will apply internal changes less often than the learners that are forced to hang on to the most general environment they have posited for a particular environment. A longer training period might help—recall from the previous section that this learner is extremely slow to simulate, and I was only able to complete one epoch of training for it—but I speculate that it would be much more helpful to develop a better system for deciding which rules should and should not be included in the grammar (and some such system is a necessity if we want to avoid a grammar that simply lists all the present–past pairs it has heard).

Some improvement in performance can be seen for predicting speakers' degree of preference for strong forms over weak ones. I computed, for each nonce verb, the *difference* between the rating of the pseudo-weak and the pseudo-strong form (or forms), for both Albright and Hayes's subjects and all three of our learners. I then took correlations as before. The results are shown in Table 4.4.

Model	Weak − Strong
DL	0.59, $p < 0.001$
ML	0.61, $p < 0.001$
EL	0.56, $p < 0.001$

Table 4.4: Comparison of the difference between judgments of weak and strong nonce past tenses, measured by Pearson's *r*—three versions of our learner using NSFC to give judgments versus native English speakers.

All three learners show improvement, to around the same level; it is not entirely clear why DL shows such poor results on the other tasks.

What does all of this show? Albright and Hayes (2003) tried to compare different ways of mimicking human past tense judgments. The claim here is that our model, using the NSFC system to make predictions, is a reasonable theory of how humans make these judgments. Surely relevant, therefore, is how the model compares to the systems tested by Albright and Hayes. The correlations of the outputs of those models with the human data is shown in Table 4.5.[4]

Model	Weak	Strong	Weak − Strong
Rule Model	0.76, $p < 0.001$	0.56, $p < 0.001$	0.77, $p < 0.001$
Analogy Model	0.46, $p < 0.01$	0.51, $p < 0.001$	0.63, $p < 0.001$

Table 4.5: Comparison of judgments on weak and strong nonce past tenses, and the difference between judgments of weak and strong nonce past tenses, using Pearson's *r*—two models tested in a study by Albright and Hayes, versus the native English speakers from that same study.

Our NSFC model appears to be capturing as much of the human pattern as Albright and Hayes's analogy model, but their rule-based model captures substantially more. Given that even the system without duplicate rules performs fairly well, we might think that all that is needed to reach this level of performance is to register the possibility of the various alternatives—that is,

[4] Astute readers will notice that the figures are not the same ones given by Albright and Hayes; they might also have noticed that I tested only 90 nonce verbs instead of the 92 used by Albright and Hayes. This is due to a glitch in the simulation software that meant that two verbs were mistakenly not tested. The figures here are different because I have recomputed the correlations for only the 90 verbs I tested.

to remember having seen forms following the *spling–splung* pattern and determine whether the new form matches the attested environments—and to distribute the scores over the alternatives in some way—for that is what the systems without duplicate rules do. Their output for strong nonce forms is almost all either zeroes or scores exactly equal to those of the weak nonce forms, meaning that they are distributing the scores evenly over the alternatives. It appears that DL, where the NSFC system should be most advantageous, does little better. A pessimistic view of the system suggests that performance at about this level should be taken as a baseline, and thus the analogy model tested by Albright and Hayes should be taken to be quite poor. If NSFC does not give us much, why does the rule-based model of Albright and Hayes 2003 do so much better? This is an open question, and it is an important step to explaining this kind of phenomenon.

4.3 Errors: Comparison to Real Learners

In this section I will evaluate the learner's performance as it compares to real learners' performance. Recall from Chapter 2 that the two most common types of past-tense errors on strong forms are simple erroneous *-d* marking (*ring–ringed*) and double-marking, meaning a correct internal change plus an erroneous *-d* (*ring–ranged*). There are probably also many (largely invisible) instances of non-marking (*ring–ring*). Other kinds of errors (including misapplication of internal changes, leading to *bring–brang* type errors) are comparatively rare (Xu and Pinker 1995).

I classified the predictions made by all three learners during training (not during testing) over the first fifteen thousand training cases; to speed up this process, I classified only every fiftieth prediction. The learners all learned quite quickly, but this nevertheless left a fair number of errors to classify. The results of this classification are shown in Table 4.6.[5]

I have little to say about the learner's performance on weak verbs, except that it is, as we have already seen, much easier for the learner without duplicate rules to get them right if it has already acquired the phonotactic rules responsible for the allomorphy in the *-d* suffix.

On strong verbs we see that non-marking errors predominate. That a verb should be included in the list associated with the null suffix is evidently easier to learn than its association with its

[5] For the purposes of comparison with real learners, cases in which the learner failed to apply an suffix, but the verb was supposed to take a null suffix and was otherwise correct, were considered correct—unlike in the learner—on the grounds that such errors are clearly not discernible in the child data even if they occur.

	Weak			**Strong**				
Model	Correct	Unmarked	Other	Correct	Unmarked	-*d*	Change + -*d*	Other
ML	86%	13%	1%	75%	17%	5%	1%	2%
DL	97%	2.5%	0.5%	76%	12%	11%	1%	0%
EL	99%	1%	0%	80%	12.5%	7%	0%	0.5%

Table 4.6: Survey of the errors made by the learner.

internal change. Whether children really do mean past tense a substantial part of the time we hear them use strong verbs in the present is an empirical question, one which I am not aware of any study addressing (it would need to be elicitation). The remainder are mostly the expected types of errors, largely erroneous -*d* errors, but with the occasional change-plus-*d* type error (DL also occasionally leaves a suffix off a strong verb, resulting in *kep* for *kept*—these make up all the *other* errors for this learner).

The errors are thus realistic—but are the figures realistic? Two questions come to mind. First, does the relation between error rate and input frequency discussed in Chapter 2 still hold? Second, does the learner show a realistic proportion of *ringed*-type to *ranged*-type errors? I collected error data for each strong verb for all three learners. To answer the first question, I collected the proportion of the learner's responses which were correct, excluding *non-marking* and *other* errors, as the quantity to be predicted, so that the result would be comparable to the tests carried out on real child data by Marcus *et al.* (1992) and above in Chapter 2 (though I did not look for the free-rider effect). Doing this test on real child data always shows a statistically significant relation between correct usage rate and frequency. Collecting these statistics revealed that the data I collected about the learner's errors would not permit a fine-grained test of any relation with frequency—when divided up verb by verb, the total number of predictions the learner made for each was very small—usually one or two. Almost all of the correct usage rates, therefore, were either zero or one. This prevents us from testing for a particular relation between frequency and correct usage rate.

Getting more data to solve the problem by simply looking at more of the learner's predictions would be difficult, since the learner only makes as many predictions as it has seen examples for a particular verb. The right way to get more data would be to test the learner on each verb more than once after each epoch, something I did not do.

We can only use our data for a crude test of the hypothesis that there is a relation between frequency and correct usage rate. For each learner, I divided up the verbs into two sets: the ones that learner failed on, and the ones that learner performed perfectly on. I then tested the hypothesis that the average (mean) frequency of the perfect verbs was higher than the average frequency of the verbs the learner failed completely on; the difference in the means proved to be statistically significant, and in the expected direction (better performance on more frequent verbs) for all three learners.[6] This shows not only that there is good evidence for a relation between frequency and correct usage rate for our learner, just as there is for real children, but also that our data, although it is very sparse, is fairly representative of our learner's performance, because, if it were not, we would probably not expect to see a clear relation to frequency.

The second question we posed above was whether the learner showed a realistic relation between the number of erroneous *-d* errors and the number of change-plus-erroneous *-d* errors. This is a real concern, since the learner acquires its knowledge of which suffix should apply to a particular strong verb *independently* of its knowledge of whether an internal change should apply. If real learners do not do this, we might expect them to have far fewer errors in which knowledge of the internal change is clearly dissociated from knowledge of the correct suffix, compared to the learner. As noted above, when our learner produces unmarked stems (we assume, present tense forms) instead of past-tense forms, it is doing just this—but, as discussed, I know of no test in the literature attempting to ascertain whether children make these errors. *Ranged*-type errors might constitute another case of learners showing a dissociation between (correct) internal-change knowledge and (incorrect) suffix knowledge (though we can come up with alternate theories, as researchers assuming the dual-mechanism hypothesis were forced to do—see fn 5)—but just looking at the data above should allay our fears that the learner makes too many such mistakes—it makes few compared to the number of *ringed*-type errors it makes, just as real learners do (see the Appendices in Marcus *et al.* 1992).

One pattern the learner does not show is the pattern often called the "U-shaped" curve. It does have fewer errors after more training (the right-hand side of the U), but it does not show an early stage with excellent performance on strong verbs, followed by a steep decline. Nevertheless, an early stage with excellent performance on strong verbs, followed by a steep decline, *is* attested in human learners. It has become entirely uncontroversial for neural-network modellers

[6] For ML, a one-sided t-test showed a difference in the means with $p < .00001$; for DL and for EL, the same test showed a difference in the means with $p < 0.001$.

working on this problem to arrange complicated “training schedules,” whereby different verbs are introduced at different times, in the hopes of duplicating the effect (for example, Rumelhart and McClelland 1986, Plunkett and Marchman 1993, Joanisse and Seidenberg 1999).

We could perhaps find some reason to think of the fact that our learner does not go through such a stage as a *good* thing. After all, as MacWhinney and Leinbach (1991) point out, the pattern is not really as robust as some researchers may imagine (see Marcus *et al.* 1992), and only holds of certain verbs. The explanation for it is not obvious, and it is thus not clear that our model *should* account for it, even to the extent that it does hold, because it might not be related to the past tense mapping per se. For example, suppose that there is a period previous to which the child simply does not understand tense marking at all. The supposedly perfect past-tense marking in the first part of the “U” would then simply represent a failure to distinguish present from past.[7] The change would then be a morphosyntactic fact, far outside the scope of the current model. Nevertheless, although its origin remains elusive, the U-shaped pattern is attested to some degree in various domains, and can be duplicated without resort to training schedules under a model which assumes, as here, conservative learning, but with slower generalization and more reliable early, pre-generalized mappings (see, for example, Alishahi 2008). The current model sheds no light on the problem.

Apart from this, however, the assessment is somewhat optimistic: the learner we have developed shows behaviour that looks roughly like what English learners do—but the facts on the acquisition of the English past tense are fairly rough themselves. Ideally, a model can give us new predictions to test. The one this learner seems to be giving us is that the most common sort of past-tense marking error in children, at least on strong verbs, ought to be no marking at all. If this prediction could be shown to be false, we would likely conclude that the learner in its current state is incorrect.

4.4 Contribution

This exploration of phonological learning began with the goal of investigating whether Minimal Generalization Learning and some stochastic system like Variational Learning can be made to work together to describe the acquisition of morphophonological grammars under reasonably sophisticated theoretical assumptions. The answer seems to be *yes*: although we have

[7] An elicitation study would be able to probe this, but none has been done to my knowledge.

constructed a fairly complex learning system, it is essentially based on these two ideas. The results described in this chapter suggest mixed prospects for the current system: our stochastic system is able to sort out morpholexical marking quite easily for the purposes of deducing a reasonable grammar, following a course of development very roughly like that seen in children, but this use of MGL to model native speakers' variable judgments of possible forms is less accurate than the MGL-based system developed in Albright and Hayes 2003.

Our experimentation with the learner also confirms that the problem of assigning blame is one to be taken seriously—that is, a system learning grammars with multiple independent parts needs prediction error from each of these parts taken individually. For our learner, that means failure if the information that it gets to update its internal-change system is not clearly distinguished from the information it gets to update its suffixation system. Thus, while Yang's (2002) suggestion that we might be able to finesse the credit–blame problem was insightful, it is clearly not viable for this kind of learning problem.

This raises an important issue, however, that we have left unresolved. Recall that our learner was never able to find a general rule for all three allomorphs of the *-d* past tense, because it was only intended to simulate the beginning of one cycle of affixation: the application of rules to individual stems and the selection of an appropriate past-tense morpheme. We assume a model of morphology in which rules can apply after this, so that rules adjusting the [d] allomorph to [ɪd] or [t] after are impossible. While it is straightforward and plausible that the learner can cleanly separate the domain of suffixation (insertions at the right edge) from the domain of internal changes (other changes), it is less clear how the learner could distinguish between internal changes and changes taking place after affixation.

One thing we have not taken into account here is the role of morphological conditioning: all the internal changes seen in the English past tense are restricted, naturally, to the past tense. The learner could perhaps be made to submit only morphologically restricted changes to be incorporated into the internal-change system, leaving all other non-suffixation errors for phonology more generally. How would the learner know that the internal changes in the past tense are morphologically restricted? Perhaps the answer is simple—we have been assuming that the learner already knows what a verb's stem is before it tries to learn its past tense. The learner can then be fairly certain that the internal changes seen in the past tense *do not occur on the surface* in the stem. They can thus be relegated straightforwardly to the internal-change system—but why would the learner not then make the insertion of [ɪ] an internal change? Such a grammar might be possible, but we would also like to somehow influence the learner to take into account phonotactics, and assign phonotactically-motivated rules to the post-affixation phonology. How this might be done is still unclear.

Of course, even considering what our learner *can* do, it does not do it all perfectly: recall that the suffix-extraction algorithm we have been assuming uses the rule, *Is the previous string of segments the same as the last few segments of the stem?* to determine where the suffix ends. This is clearly a stand-in for a more sophisticated algorithm, since this algorithm would posit *-erstood* as a past-tense suffix for *understand*—wrongly predicting that children should make errors like *understanderstood*. Above, we remedied this by simply restricting the length of suffixes, which is clearly not a solution in general. Furthermore, our segmentation system assumes that the learner knows something about all but one of the morphemes in the word—but how do learners come to their existing knowledge of *any* morphemes' phonological forms in the general case, in which none of the morphemes in a word ever occur in isolation?

These issues, along with various others I have raised throughout the text, surely yield no shortage of modelling experiments—segmentation, deduction of underlying forms, recognition of underlying forms in the input, the acquisition of rule ordering, the acquisition of long-distance patterns, the correct characterization of human phonological generalizations—for researchers wanting to follow the same general path as the current study. Resolving any of these issues to any degree of satisfaction would help to improve the current system too. Attempting to address these other issues using other data is surely the right approach to improving the current system: only so much can be done with the English past tense. Our system for deducing English past tenses is nevertheless novel in its approach, motivated far more strongly by linguistic theory than any previous learner.

I have also presented a negative result in this paper—in section 2.2, where I rejected Yang's (2002) claims of a "free-rider effect" in the acquisition of English strong past tenses, an effect Yang claimed demonstrated the necessity of a rule-based, stem-marking model of morphophonology. The viability of the current learner shows that the failure of the free-rider effect in no way weakens the case for such a theory of morphophonology. It is my hope that—with or without these arguments against the simplistic dual-mechanism theories of morphological processing advanced during the previous decade—the general approach I have taken to developing an automatic learner grounded in linguistic theory will stimulate a more nuanced approach to the psycholinguistic issues in morphology, and to the study of language acquisition generally.

Appendix A

Detailed Proofs and a Crash Course in Probability Theory

In the text, I have omitted the proof of certain results relating to learning in the L_{RP} scheme. These are included here. These proofs require only basic probability theory, which I have attempted to explain below.

Basic predictions of a simple L_{RP} Variational Learner.

Suppose we have a linguistic lesson with two candidate grammoids, G_1 and G_2, with penalty probabilities c_1 and c_2 respectively. Recall from the text that the *penalty probability* of a grammoid is a piece of information about the learner's input: it is the probability with which the learner will hear an utterance for which that grammoid is not "fit" (thus, which will cause it to be penalized if it is chosen). The goal of this section is to get some idea what we should expect an L_{RP} Variational Learner's belief strengths to be after t steps of learning. In particular, we would like to know how this relates to c_1 and c_2.

We start by assuming that the penalty probabilities do not change over time: that is, that the relative frequency of various types of utterances in the child's input does not change over time, and (more importantly) the learner's ability to process these utterances for the purposes of learning does not change. This latter assumption is a simplifying assumption, but together these assumptions mean we can find the relative frequency of whatever sort of evidence we think is relevant to penalizing the grammoid in question from some corpus and take that figure to be the penalty probability. We formalize this by saying that the input at time t is not known with certainty; rather, it is a *random variable*—a set of possible events $\{e_1, e_2, \dots\}$ each with an associated *probability*—a real number on $[0,1]$—with the probability of e written $Pr(e)$. These probabilities together form a *probability distribution*, upon which we impose the restriction that the sum of all probabilities must be one (if we add two probabilities together, the total should be the probability of *either* of the two events occurring, and we want one to mean total certainty, so this follows from the assumption that *something* must happen). We do not know the full distribution of the various kinds of input—we do not need to, since the probabilities c_1 and c_2 are the only ones that are relevant—but our assumption is that the distribution is fixed for all t.

Once we treat the learner's input as a random variable, the values of $B_t(G_1)$ and $B_t(G_2)$, being functions of the learner's input, must be treated as random variables themselves. Belief strengths thus also have probability distributions. This might be somewhat confusing: after all, belief strengths *are* probability distributions—but this is not the probability distribution we are talking about. Rather, we are talking about the distribution *of this distribution.*[1]

Unlike the input, however, the distribution of the learner's belief strengths for some grammoid is not fixed over time: belief strengths at some time t are dependent not only on the input at t, but also on the learner's beliefs at the previous time $t-1$, so their distributions are not fixed for all t. Of course—if a learner's belief in G_1 were as likely to have strength 0.5 after one time step as after one thousand, there would be nothing being learned.

[1] It might help to consider the two standard interpretations of what a probability distribution "means." One is the *frequentist* interpretation—probabilities are just the relative frequencies of events. If we were thinking as frequentists, we could recast the learner's belief strength for some grammoid as just a measurement of how often the learner will choose that grammoid, for learning or for production, relative to other grammoids. The other is the *Bayesian* interpretation—as a degree of belief. This is how we are interpreting the learner's belief strengths already. Many mathematicians and philosophers have debated just which interpretation is "correct," but my own feeling is that we can better interpret the learner's internal probabilities with a Bayesian interpretation, and better understand what *the probability distribution of a probability distribution* might be if we have a frequentist interpretation of the probabilities *of* these internal probabilities. This interpretation goes like this: the learner's belief strength in G_i at some time t, $B_t(G_i)$, is a random variable, so that $B_t(G_i)$ is only (say) 0.72 with some probability $Pr(B_t(G_i) = 0.72)$, then we will interpret that to mean that only a certain fraction of runs of the learner will have belief strength 0.72 in G_i at time t—just as only a certain fraction of the inputs we expect to turn up at time t will be ones that G_i is unfit for. To keep the reader's mental picture clear, I use the term "beliefs" for probability distributions that I want the reader to think of in a Bayesian way—the things that are inside the learner's mind—and "probabilities" for distributions I want the reader to think of in a frequentist way—the things that are out in the world (including various learners' minds—as seen from the outside). (I could instead have chosen to think of both kinds of probabilities in a Bayesian way. That would mean thinking of the distribution of belief strengths as the strength of *our belief* in the fact that the learner has such-and-such a belief strength for some grammoid at some time. It is certainly too confusing to *talk* about these probabilities in that way, but the reader is free to think of them that way.)

Recall: the goal of this section is to predict the learner's belief strengths at time t—so what are we trying to predict if belief strengths are mere random variables, with no certain value? We can take an average, formalized by *expected value* or *expectation*, given in (50), a function of some random variable X. (It is standard notation to write any of the possible events for a random variable X as x—so here we sum over the possible events of X—and to write the probability of each of those events as $Pr(X = x)$, or just $Pr(x)$.)

(50) $$E[X] \quad = \quad \sum_x Pr(x) \cdot x$$

It makes sense to talk about the expected value of a random variable when each of the possible "events" is *the random variable's having some particular numeric value*. The expected value of $B_t(G_i)$, $E[B_t(G_i)]$, thus tells us what belief strength G_i will have at time t for the *average* run of a learner in some linguistic environment with the given input distribution (although, of course, just as there is no real "average Canadian family" with 1.1 children at home, "the average run of a learner" may not really ever occur). What we want is a formula for $E[B_t(G_i)]$ (we don't care what t is) in terms of c_i.

In order to do this, we need to be able to enumerate what each of the various "events" is, and set down a formula for expected value that contains the probability of each. This sounds hard—the trick is to develop a formula for expected value of *the difference between* $B_t(G_i)$ *and* $B_{t+1}(G_i)$, and then sum these up: *the expected value of the sum of several quantities*—in this case, the initial belief strength and all the increases and decreases after that—can be shown to be equal to *the sum of the expected value of each of those quantities*. Once we have a formula for the expected value of the change in the belief strength for G_i at some arbitrary time t, we can find the expected value of the belief strength itself at whatever time time we like by summing up the expected values of the change in belief strength. And we know how to find the expected value of the change in a grammoid's belief strength, because we know how grammoids change—they change by the L_{RP} scheme. The possible events are the possible belief-strength updates, of which there are four. The L_{RP} scheme is repeated in (51).

(51)
$$
\begin{array}{ll}
\text{Grammoid } G_i \text{ derives input:} & \begin{cases} B_{t+1}(G_i) = B_t(G_i) + \gamma \cdot (1 - B_t(G_i)) \\ (\textbf{Reward} \text{ to } G_i) \\ B_{t+1}(G_j) = B_t(G_j) - \gamma \cdot B_t(G_j) \ \ (j \neq i) \\ (\textbf{Decrements} \text{ to all } G_{j \neq i}) \end{cases} \\
\text{Otherwise:} & \begin{cases} B_{t+1}(G_i) = B_t(G_i) - \gamma \cdot B_t(G_i) \\ (\textbf{Penalty} \text{ to } G_i) \\ B_{t+1}(G_j) = B_t(G_j) + \gamma \cdot (\frac{1}{N-1} - B_t(G_j)) \ \ (j \neq i) \\ N = \text{total number of grammoids} \\ (\textbf{Compensation} \text{ to all } G_{j \neq i}) \end{cases}
\end{array}
$$

Under the L_{RP} scheme, the learner *either* chooses G_i for its simulated production at time t or it does not; then, it *either* increases G_i's belief strength or decreases it. Here are the relevant events, then, at time t: first, the learner might choose G_i and *reward* it (add $\gamma \cdot (1 - B_t(G_i))$ to its old belief strength, $B_t(G_i)$); on the other hand, the learner might not choose G_i, but find that the alternative grammoid it chose was fit—causing it to *decrement* the grammoids not chosen, including G_i, to make up for its increase in belief in the chosen alternative (that is, it would subtract $\gamma \cdot B_t(G_i)$ from G_i's old belief strength, $B_t(G_i)$); the learner might also remove belief strength from G_i if it chose G_i and found it unfit—if it *penalized* G_i (again subtracting $\gamma \cdot B_t(G_i)$ from G_i's old belief strength, $B_t(G_i)$); finally, the learner might also add belief strength to G_i if it chose some other grammoid but found it unfit—it would need to penalize that other grammoid, and it would thus *compensate* the other grammoids, including G_i, to keep the belief strengths summing to one (by adding $\gamma \cdot (\frac{1}{N-1} - B_t(G_i))$ to G_i's old belief strength, $B_t(G_i)$). These are the only four possible changes in the belief in G_i, and thus these are the four events. We thus need only find the probabilities of each of these events to have a formula for the expected value of the change in G_i's belief strength. That formula would have the general shape in (52).

(52)
$$
\begin{array}{lcl}
\text{Prob. of reward} & \times & \gamma(1 - B_t(G_i)) \\
+ \text{ Prob. of decrement} & \times & (-\gamma)B_t(G_i) \\
+ \text{ Prob. of penalty} & \times & (-\gamma)B_t(G_i) \\
+ \text{ Prob. of compensation} & \times & \gamma(\dfrac{1}{N-1(=1)} - B_t(G_i))
\end{array}
$$

If we filled in the probabilities here, this formula would give us an average for the change in the belief strength of G_i between some time t and $t+1$—but it would be in terms of $B_t(G_i)$, which, like $B_{t+1}(G_i)$, is a random variable. Recall that our goal was to find a formula for $E[B_{t+1}(G_i)]$ by summing together t formulas for the change in the belief in G_i—we want that formula to give us a constant, not a random variable or a function of random variables. We can do this if the formula for the expected value of the change in belief strength is always in terms of constants rather than random variables.

Note, however, that, while $B_t(G_i)$ is a random variable, $E[B_t(G_i)]$ is a constant—and, as it turns out, it is easy to get the above formula in terms of $E[B_t(G_i)]$ rather than $B_t(G_i)$.

Let us begin by giving a name to the formula we are trying to derive. Write the input at time t as U_t, and write the change in a belief strength b given some input u under the L_{RP} scheme as $\Delta(b,u)$. Then we are looking for a formula for $E[\Delta(B_t(G_i),U_t)]$.

To get a formula for $E[\Delta(B_t(G_i),U_t)]$ in terms of $E[B_t(G_i)]$, we must first introduce the notion of *conditional expectation,* $E[X|Y=y]$:

$$E[X|Y=y] \quad = \quad \textstyle\sum_x Pr(X=x|Y=y)\cdot x \tag{53}$$

The expected value of X given that Y is y, or $E[X|Y=y]$, is the expected value of X *if we know* that $Y=y$. To understand this we need to understand *conditional probability*. A conditional probability distribution is a probability distribution like any other, except that it gives the probability of seeing each value of a random variable once we know the value of some second random variable. If, for example, we did not know for certain the location of some fugitive, or whether it was raining there, these would be two random variables (say, L for location and R for whether it is raining there), each with some probability distribution. Suppose, however, that it was established (beforehand) that it never rained in New Belgrade; if, hypothetically, we were to discover that the fugitive was hiding in New Belgrade, we would know that it was not raining where the fugitive was; $Pr(R = \text{True}|L = \text{New Belgrade}) = 0$, $Pr(R = \text{False}|L = \text{New Belgrade}) = 1$ makes up this *conditional* probability distribution. The conditional expected value of a random variable is just its expected value when calculated against a conditional probability distribution.

It is important to distinguish $E[X|Y=y]$ from $E[X|Y]$ (which will become useful below). $E[X|Y=y]$ is a constant, which we can think of as the (hypothetical) average value of X when we know that $Y=y$. $E[X|Y]$, on the other hand, is the function of the random variable Y

which tells us what the conditional expected value of X is for arbitrary Y. Functions of random variables, rather than of particular values of those random variables, are, naturally, random variables themselves. $E[X|Y]$ is a random variable with its own probability distribution: the probability that $E[X|Y]$ is equal to $E[X|Y=y]$ is equal to the probability that $Y=y$.

We can get a formula for $E[\Delta(B_t(G_i),U_t)]$ if we know the *law of total expectation*. The law of total expectation states that the expected value of a random variable X is equal to the expected value *of its expected value, conditional on some other random variable*. This is stated (without proof) in (54).

$$E[X] \quad = \quad E[E[X|Y]] \tag{54}$$

The upshot of (54) for us is that $E[\Delta(B_t(G_i),U_t)] = E[E[\Delta(B_t(G_i),U_t)]|B_t(G_i)]]$ (functions of random variables are random variables themselves, so $\Delta(B_t(G_i),U_t)$ is a random variable, and has an expected value). We can thus proceed as before, and get a better answer.

$$\begin{aligned} E[\Delta(B_t(G_i),U_t)] &= E[E[\Delta(B_t(G_i),U_t)]|B_t(G_i)]] \\ & \quad \text{which is, by definition,} \\ &= \textstyle\sum_b Pr(B_t(G_i)=b)\cdot E[\Delta(B_t(G_i),U_t)]|B_t(G_i)=b] \end{aligned} \tag{55}$$

To get a formula for $E[\Delta(B_t(G_i),U_t)]|B_t(G_i)=b]$, we can proceed as before:

$$\begin{aligned} & E[\Delta(B_t(G_i),U_t)]|B_t(G_i)=b] \\ = \; & Pr(\Delta(B_t(G_i),U_t) \text{ is a reward}|B_t(G_i)=b)\times\gamma(1-b) \\ + \; & Pr(\Delta(B_t(G_i),U_t) \text{ is a decrement}|B_t(G_i)=b)\times(-\gamma)b \\ + \; & Pr(\Delta(B_t(G_i),U_t) \text{ is a penalty}|B_t(G_i)=b)\times(-\gamma)b \\ + \; & Pr(\Delta(B_t(G_i),U_t) \text{ is compensation}|B_t(G_i)=b)\times\gamma(1-b) \end{aligned} \tag{56}$$

What are the conditional probabilities? They depend in part on the learner's choice of grammoid, which depends directly on the learner's belief strengths at time t: with probability $B_t(G_i)$, (which, in this case, we know to be b), G_i will be chosen, thus rewarded or penalized; with probability $1-B_t(G_i)$, (here, $1-b$), it will not be chosen, thus it will suffer a decrement or get compensation. The probabilities also depend on the input: if G_i is chosen, it will be penalized with probability c_i, rewarded with probability $1-c_i$. Otherwise, it will be compensated with probability c_j, decremented with probability $1-c_j$. In fact, we can make one further assumption here that will make our task easier: let us suppose that G_i is the correct grammoid, the one (hypothetical, of course!) for which a learner would have totally consistent input. That means that we can take c_i to be 0.

We can now fill in the conditional probabilities in (56), using the basic rule of probability that the probability of two independent events A and B *both* occurring is $Pr(A) \times Pr(B)$. This means that the conditional probability of a reward to G_i is $b \times (1 - c_i) = b$; the conditional probability of decrementing G_i is $(1-b) \times (1-c_j)$; the conditional probability of a penalty to G_i is $b \times c_i = 0$; and the conditional probability of compensating G_i is $(1-b) \times c_j$. This gives us an expression for $E[\Delta(B_t(G_i), U_t)]|B_t(G_i) = b]$ that we can simplify as in (57).

(57)
$$\begin{array}{rcl} E[\Delta(B_t(G_i),U_t)]|B_t(G_i) = b] & = & b \times \gamma(1-b) \\ & + & (1-b)(1-c_j) \times (-\gamma)b \\ & + & (1-b)c_j \times \gamma(1-b) \\ E[\Delta(B_t(G_i),U_t)]|B_t(G_i) = b] & = & \gamma c_j(1-b) \end{array}$$

Substituting this into our previous formula for $E[\Delta(B_t(G_i), U_t)]$ $(= E[E[\Delta(B_t(G_i), U_t)]|B_t(G_i)]])$, we get (58).

(58)
$$\begin{array}{rcl} E[\Delta(B_t(G_i),U_t)] & = & E[E[\Delta(B_t(G_i),U_t)]|B_t(G_i)]] \\ & = & \sum_b Pr(B_t(G_i) = b) \cdot E[\Delta(B_t(G_i),U_t)]|B_t(G_i) = b] \\ & = & \sum_b Pr(B_t(G_i) = b) \cdot \gamma c_j(1-b) \\ & = & E[\gamma c_j(1 - B_t(G_i))] \\ & & \text{which gives, by linearity of } E[\cdot], \\ E[\Delta(B_t(G_i),U_t)] & = & \gamma c_j(1 - E[B_t(G_i)]) \end{array}$$

In (58), we have a formula for the average change in the belief in G_i at some arbitrary time, put in terms of constants—though, admittedly, one constant that we don't know yet, $E[B_t(G_i)]$ (but we will soon solve that), and one constant we do not plan to ever know, γ (but we don't really care what γ is, as long as it does not change). We derived this formula using the fact that the expectation operator is linear (that is, that $E[X+Y] = E[X] + E[Y]$ and $E[aX] = aE[X]$, where a is not a random variable). We can use the same fact to get a formula for $E[B_{t+1}(G_i)]$, given that $B_{t+1}(G_i) = B_t(G_i) + \Delta(B_t(G_i), U_t)$ by definition.

(59)
$$\begin{aligned}
E[B_{t+1}(G_i)] &= E[B_t(G_i)+\Delta(B_t(G_i),U_t)] \\
&\quad \text{which is, by linearity of } E[\cdot] \\
&= E[B_t(G_i)]+E[\Delta(B_t(G_i),U_t)] \\
&\quad \text{which is, substituting our previous result} \\
&= E[B_t(G_i)]+\gamma\cdot c_j\cdot(1-E[B_t(G_i)]) \\
&\quad \text{giving us, after some rearranging,} \\
E[B_{t+1}(G_i)] &= \gamma\cdot c_j+(1-\gamma\cdot c_j)\cdot E[B_t(G_i)]
\end{aligned}$$

The equation in (59) gives us a formula for $E[B_{t+1}(G_i)]$ in terms of $E[B_t(G_i)]$. This is a *recurrence relation*: it is an expression that gives a value to every term in some sequence in terms of the previous value. In this case, it can tell us what $E[B_t(G_i)]$ ought to be for $t=1$, $t=2$, $t=3$, and so on, if we start at $t=0$ and work our way up. If we want to do this, we can start by assuming that $B_0(G_i)$ is some arbitrary constant; $E[B_0(G_i)]$ is $B_0(G_i)$, because the expected value of a constant (non-random-variable) is just that constant. We will then quickly find that $E[B_1(G_i)]$ is $\gamma c_j+(1-\gamma c_j)B_0(G_i)$, which can be rearranged, if we are prescient (we will soon see why), to $1-(1-\gamma c_j)(1-B_0(G_i))$. We can then substitute this again—we get $\gamma c_j+(1-\gamma c_j)[\gamma c_j+(1-\gamma c_j)B_0(G_i)]$ for $E[B_2(G_i)]$, which can be simplified to $1-(1-\gamma c_j)^2(1-B_0(G_i))$—and so on. In short, armed with the formula in (59), if we want to find some value of $E[B_t(G_i)]$, we can simply apply (59) t times to find it. We thus have a simple expression of the entire sequence.

If we are prescient, however, we can do one better. It is not practical to apply (59) t times in general, let alone to try and simplify the large expressions that result for even reasonably small values of t. To avoid this, the prescient mathematician will attempt to find a *closed-form solution* to the recurrence relation. This means a general formula for $E[B_t(G_i)]$ that allows us to calculate it for arbitrary t without working out all the values for smaller t. As hinted at, there is such a solution. It is shown in (60).

(60)
$$E[B_t(G_i)] \;=\; 1-(1-\gamma c_j)^t(1-B_0(G_i))$$

Finding a closed-form solution to a recurrence relation is often the result of a lucky guess, but, once we make a guess, we can prove the solution correct by mathematical induction. We begin by showing that the solution would be right if the only value of t we were considering were $t=0$. This is easy:

$$
\begin{aligned}
1-(1-\gamma c_j)^0(1-B_0(G_i)) &= 1-1\cdot(1-B_0(G_i)) \\
&= 1-1+B_0(G_i) \\
&= B_0(G_i)=E[B_0(G_i)]
\end{aligned}
\tag{61}
$$

(61) shows that, restricting our consideration to $t=0$, the "generalization" in (60) is a true one (because, as we pointed out, the expected value of a constant is just that constant). Mathematical induction means then attempting to prove that, *if* the generalization holds for all t up to and including T, for arbitrary T, *then* it holds for $t=T+1$. If we already know that it holds for $t=0$, then we have thus proven not only that it holds for $t=1$—and thus, t up to and including 1—but, therefore, that it holds for $t=2$, and thus, t up to and including 2—and thus that it holds for t up to and including 3, and thus that it holds for t up to and including 4, and so on, like dominoes, *ad infinitum*. Here, we can easily prove that if $E[B_t(G_i)]=1-(1-\gamma c_j)^t(1-B_0(G_i))$ for all $t\leq T$, then $E[B_{T+1}(G_i)]=1-(1-\gamma c_j)^{T+1}(1-B_0(G_i))$.

The recurrence relation says that

$$
\begin{aligned}
& E[B_{T+1}(G_i)] \\
= \;& \gamma c_j+(1-\gamma c_j)E[B_T(G_i)]
\end{aligned}
$$

and so, if we know that the solution
is correct for values of t up to T, we have

$$
= \gamma c_j+(1-\gamma c_j)\cdot[1-(1-\gamma c_j)^T(1-B_0(G_i))] \tag{62}
$$

which is, simplifying,

$$
\begin{aligned}
&= \gamma c_j+(1-\gamma c_j)-(1-\gamma c_j)(1-\gamma c_j)^T(1-B_0(G_i)) \\
&= 1-\gamma c_j+\gamma c_j-(1-\gamma c_j)(1-\gamma c_j)^T(1-B_0(G_i)) \\
&= 1-(1-\gamma c_j)(1-\gamma c_j)^T(1-B_0(G_i)) \\
&= 1-(1-\gamma c_j)^{T+1}(1-B_0(G_i))\square
\end{aligned}
$$

By mathematical induction, we have proven that, in general, $E[B_t(G_i)]=1-(1-\gamma c_j)^t(1-B_0(G_i))$. We can use this to make predictions about a child's behaviour at time t—even if we do not know the exact value of t, (and we cannot be expected to know what sort of scale t should be on, even if we know how old the child is), or of γ or $B_0(G_i)$ (and we surely cannot be expected to know what either of these things really are). These are all arbitrary constants—we don't care what they are (though if we wanted to examine the child's progress over time we might care about comparing larger to smaller values of t), so for clarity, we can rewrite the formula as $1 - K(1 - Lc_j)^t$—and if we make the simplifying assumption that that the child's knowledge is not changing over the period we are looking at, we can derive some simple predictions from such a formula.

Prediction of the inflection-learning model in Yang 2002.

This section assumes that the reader is familiar with the basic concepts from probability introduced in the previous section, and with the idea that we can find a formula for the learner's knowledge at time t as an *expected value*—an average over all the possible sequences of inputs and stochastic choices—and that we can do this by finding a solution to a *recurrence relation*—a statement of the learner's knowledge at some time step in terms of its knowledge at the previous time step.

In this section, we will use these techniques to find a formula for the knowledge of a learner acquiring past-tense markings for English verbs—having already found the set of rules that inflect English verbs, and determined which are morpholexical (see section 1.4)—behaving as in (63).

(63) On processing a past tense verb form X with stem x, to determine whether rule R gives X...

a. Stochastically choose a marking grammoid for x—either allowing or disallowing R's application..

b. Stochastically choose a grammoid telling us whether R does or does not apply at all (or whether it really "exists").

c. Apply/don't apply R to x (falling back on *-d* if we don't apply R):

(i) If the output matches X, reward our beliefs in both x's marking grammoid *and* R's existence grammoid.

(ii) If the output does not match X, penalize our beliefs in both x's marking grammoid *and* R's existence grammoid.

We will attempt to get a formula for the learner's knowledge in terms of input frequencies, or, what amounts to the same thing, the probability of the learner hearing one of the relevant inputs. We assume that the relevant inputs are the past tenses of various verbs. In particular, the probability of hearing a verb v at time t can be measured by the fraction of the utterances the child is exposed to that contain v. We call this quantity f_v. To get Yang's *free-rider effect*, we also take the probability of hearing any *other* verb inflected by the same rule to be relevant. This can be found by finding the sum of f_h for all $h \neq v$ inflected by the same rule as v. We call this quantity F_v. The goal is thus to get a formula the learner's knowledge in terms of f_v and F_v.

As we will see, this is not trivial. The final result will be a gross approximation, showing that, although very simple predictions can be made analytically (see, for example, section 1.3), Variational Learning does not have the advantage of giving us clear predictions in realistic learning scenarios. In general, we must resort to computer simulation.

We begin by noting that the quantity representing the learner's knowledge of the correct past-tense inflection of v at time $t+1$, for some v that requires association with a morpholexical rule r under Yang's theory, is the strength of belief in v being marked for r—which we wrote in the text as $B_{t+1}(\top_{v,r})$—multiplied by the strength of belief in the existence of r—which we wrote in the text as $B_{t+1}(\top_r)$. (This is because belief strengths are probabilities, and a basic rule of probability says that the probability of two independent events occurring is obtained by multiplying the two events' individual probabilities.) We write the relevant belief as $B_{t+1}(\top_{v,r})B_{t+1}(\top_r)$, so that, proceeding along the lines of the previous section, we will try to find a formula for $E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)]$.

We know from the previous section that we can obtain a formula for $E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)]$ in terms of previous beliefs by making reference to the *change* in a particular belief at time t, B_t, given the input at time t, U_t; we write this change as $\Delta(B_t, U_t)$. Here we get (64).

$$
\begin{aligned}
& E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)] \\
= \quad & E[(B_t(\top_{v,r}) + \Delta(B_t(\top_{v,r}),U_t))(B_t(\top_r) + \Delta(B_t(\top_r),U_t))]
\end{aligned}
$$

which we can expand and rewrite, by the linearity of $E[\cdot]$, as

(64)

$$
\begin{aligned}
= \quad & E[B_t(\top_{v,r})B_t(\top_r)] \\
+ \quad & E[B_t(\top_{v,r})\Delta(B_t(\top_r),U_t)] \\
+ \quad & E[B_t(\top_r)\Delta(B_t(\top_{v,r}),U_t)] \\
+ \quad & E[\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)]
\end{aligned}
$$

This formula for $E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)]$ will, as before, be a recursive statement of the sequence (a recurrence relation). One term is already in terms of a constant that we will take as given in our recurrence relation, $E[B_t(\top_{v,r})B_t(\top_r)]$; as before, however, we will need to expand the other three terms, because they are expected values of functions of $\Delta(B_t(\top_{v,r}),U_t)$, $\Delta(B_t(\top_r),U_t)$, and $\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)$, and we do not know how to compute these. We must find formulas for each of these three additional terms.

First, we attempt to find a formula for $E[B_t(\top_{v,r})\Delta(B_t(\top_r),U_t)]$, using, as in the previous section, the fact that $E[X] = E[E[X|Y]]$ (the law of total expectation), or, in general, that $E[X] = E[E[X|Y_1,\ldots,Y_n]]$. Appealing only to the definition of expectation and to basic properties of arithmetic, we have (65). (I write $Pr(X = x)$ as $Pr(x)$, $Pr(X = x \text{ and } Y = y)$ as $Pr(x,y)$, $Pr(X = x|A = a, B = b)$ as $Pr(X = x|a,b)$ to save space.)

$$
\begin{aligned}
& E[B_t(\top_{v,r})\Delta(B_t(\top_r),U_t)] \\
= \quad & E[E[B_t(\top_{v,r})\Delta(B_t(\top_r),U_t)|B_t(\top_{v,r}),B_t(\top_r)]]
\end{aligned}
$$

By the definition of $E[\cdot]$, this is

$$
= \quad \textstyle\sum_{b_v,b_r} Pr(b_v,b_r) \cdot \sum_\delta Pr(B_t(\top_{v,r})\Delta(B_t(\top_r),U_t) = b_v\delta|b_v,b_r) \cdot b_v \cdot \delta
$$

Since b_v is a constant in the inner summation, this is

$$
= \quad \textstyle\sum_{b_v,b_r} Pr(b_v,b_r) \cdot b_v \cdot \sum_\delta Pr(B_t(\top_{v,r})\Delta(B_t(\top_r),U_t) = b_v\delta|b_v,b_r) \cdot \delta \tag{65}
$$

Finally, given that $B_t(\top_{v,r}) = b_v$, the only way for $B_t(\top_{v,r}) \cdot \Delta(B_t(\top_r),U_t)$ to be equal to $b_v\delta$ is for $\Delta(B_t(\top_r),U_t)$ to be δ. Thus,

$$
\begin{aligned}
& E[B_t(\top_{v,r})\Delta(B_t(\top_r),U_t)] \\
= \quad & \textstyle\sum_{b_v,b_r} Pr(b_v,b_r) \cdot b_v \cdot \sum_\delta Pr(\Delta(B_t(\top_r),U_t) = \delta|b_v,b_r) \cdot \delta
\end{aligned}
$$

Similarly, for $E[B_t(\top_r)\Delta(B_t(\top_{v,r}),U_t)]$, we have:

$$\begin{aligned} & E[B_t(\top_r)\cdot\Delta(B_t(\top_{v,r}),U_t)] \\ = \; & \textstyle\sum_{b_v,b_r} Pr(b_v,b_r)\cdot b_r\cdot\sum_\delta Pr(\Delta(B_t(\top_{v,r}),U_t)=\delta|b_v,b_r)\cdot\delta \end{aligned} \tag{66}$$

Finally, for $E[\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)]$, we have:

$$\begin{aligned} & E[\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)] \\ = \; & \textstyle\sum_{b_v,b_r} Pr(b_v,b_r)\cdot\sum_{\delta_v\delta_r} Pr(\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)=\delta_v\delta_r|b_v,b_r)\cdot\delta_v\delta_r \end{aligned} \tag{67}$$

To put these formulas in terms of known quantities, we must examine the details of the learning model to expand $\sum_\delta Pr(\Delta(B_t(\top_r),U_t)=\delta|b_v,b_r)\cdot\delta$, $\sum_\delta Pr(\Delta(B_t(\top_{v,r}),U_t)=\delta|b_v,b_r)\cdot\delta$, and $\sum_{\delta_v\delta_r} Pr(\Delta(B_t(\top_{v,r}),U_t)\cdot\Delta(B_t(\top_r),U_t)=\delta_v\delta_r|b_v,b_r)$. Recall from the previous section that we can do this by considering, in each case, all the changes in the learner's state that are possible given the learning model, finding the probability of each (given that $B_t(\top_{v,r})=b_v$ and $B_t(\top_r)=b_r$), and multiplying by the value of the change (or product of two changes). We are still using the L_{RP} scheme for learning, which I repeat here as (68).

$$\begin{array}{ll} \text{Grammoid } G_i \text{ derives input:} & \begin{cases} B_{t+1}(G_i)=B_t(G_i)+\gamma\cdot(1-B_t(G_i)) \\ (\textbf{Reward} \text{ to } G_i) \\ B_{t+1}(G_j)=B_t(G_j)-\gamma\cdot B_t(G_j) \quad (j\neq i) \\ (\textbf{Decrements} \text{ to all } G_{j\neq i}) \end{cases} \\ \text{Otherwise:} & \begin{cases} B_{t+1}(G_i)=B_t(G_i)-\gamma\cdot B_t(G_i) \\ (\textbf{Penalty} \text{ to } G_i) \\ B_{t+1}(G_j)=B_t(G_j)+\gamma\cdot(\frac{1}{N-1}-B_t(G_j)) \quad (j\neq i) \\ N=\text{total number of grammoids} \\ (\textbf{Compensation} \text{ to all } G_{j\neq i}) \end{cases} \end{array} \tag{68}$$

The L_{RP} scheme in (68) tells us that, at some time t, the learner can change its belief in a grammoid in one of four ways; each of these ways of changing belief in G at time t corresponds to a different value of $\Delta(B_t(G),U_t)$.

In the previous section, we took these four values of $\Delta(B_t(G),U_t)$ to be exhaustive: at each time step, we supposed, the learner *must* change its belief in G in one of these four ways. Given that none of these four ways of changing belief in G corresponds to *no change*, this means we assume that the learner changes its belief in G at every time step. This is perhaps reasonable for a linguistic lesson like whether the language has *pro*-drop, which is arguably relevant to almost every utterance the child hears. If the child must internally duplicate its input, then there can be no avoiding a choice of grammoids for *pro*-drop, and thus no avoiding a belief change.

I will *not* however, take this to be the case for past tense learning: that is, I will *not* take it to be the case that the learner's belief in $\top_{v,r}$ will change on every input, or that the learner's belief in $\top_r$ will change on every input. Rather, I will suppose that the change in the learner's belief in $\top_{v,r}$ will be zero if the learner does not hear the past tense of v at time t; similarly, the change in the learner's belief in $\top_r$ will be zero if the learner does not hear the past tense of some verb inflected by rule r at time t. This seems to me to be consistent with the theory as presented in Yang 2002.

The upshot of this for us is that the conditional probabilities of rewards, decrements, penalties, and compensations will be slightly different than they were in the previous section. In the previous section, the conditional probabilities of rewards, decrements, penalties, and compensations each included two factors: one factor corresponding to the probability of choosing some grammoid at time t—which was *only* a belief strength; and a second factor corresponding to the probability of that grammoid leading to a correct or incorrect derivation—which was *only* the probability of a certain relevant type of input. Now, the factor corresponding to the probability of choosing some grammoid at time t, but that factor will need to have two factors of its own: one, a belief strength, corresponding to the probability of selecting some grammoid, as before; and a second, an input probability, corresponding to the probability of hearing an input that would trigger a choice to choose that grammoid in the first place.

This is given for $\sum_\delta Pr(\Delta(B_t(\top_r), U_t) = \delta | b_v, b_r) \cdot \delta$ in (69).

$$
\begin{aligned}
& \textstyle\sum_{\delta} Pr(\Delta(B_t(\top_r), U_t) = \delta | b_v, b_r) \cdot \delta \\
= \quad & \text{Prob. of reward to } \top_r \times \gamma(1 - b_r) \\
+ \quad & \text{Prob. of decrement to } \top_r \times (-\gamma) b_r \\
+ \quad & \text{Prob. of penalty to } \top_r \times (-\gamma) b_r \\
+ \quad & \text{Prob. of compensation to } \top_r \times \gamma(\frac{1}{N(=2)-1} - b_r) \\
(+ \quad & \text{Prob. of no change} \times 0 = 0)
\end{aligned}
$$

As before, we expand this to:

$$
\begin{aligned}
= \quad & \text{Prob. of choosing } \top_r \times \text{Prob. of correct output with } \top_r \times \gamma(1 - b_r) \\
+ \quad & \text{Prob. of choosing } \perp_r \times \text{Prob. of correct output with } \perp_r \times (-\gamma) b_r \\
+ \quad & \text{Prob. of choosing } \top_r \times \text{Prob. of wrong output with } \top_r \times (-\gamma) b_r \\
+ \quad & \text{Prob. of choosing } \perp_r \times \text{Prob. of wrong output with } \perp_r \times \gamma(1 - b_r)
\end{aligned}
$$

(69)

Here, however, we expand this further, to get

$$
\begin{aligned}
= \quad & \text{Input prob. of a verb taking } r \times b_r \times \text{Prob. of correct output with } \top_r \times \gamma(1 - b_r) \\
+ \quad & \text{Input prob. of a verb taking } r \times (1 - b_r) \times \text{Prob. of correct output with } \perp_r \times (-\gamma) b_r \\
+ \quad & \text{Input prob. of a verb taking } r \times b_r \times \text{Prob. of wrong output with } \top_r \times (-\gamma) b_r \\
+ \quad & \text{Input prob. of a verb taking } r \times (1 - b_r) \times \text{Prob. of wrong output with } \perp_r \times \gamma(1 - b_r)
\end{aligned}
$$

Finally, we can factor out the input probability and replace it with our notation:

$$
\begin{aligned}
= \quad & (f_v + F_v) \times (b_r \times \text{Prob. of correct output with } \top_r \times \gamma(1 - b_r) \\
& \qquad + (1 - b_r) \times \text{Prob. of correct output with } \perp_r \times (-\gamma) b_r \\
& \qquad + b_r \times \text{Prob. of wrong output with } \top_r \times (-\gamma) b_r \\
& \qquad + (1 - b_r) \times \text{Prob. of wrong output with } \perp_r \times \gamma(1 - b_r))
\end{aligned}
$$

We will fill in the probabilities of correct/wrong output shortly, with one exception: correct output with $\perp_r$ or $\perp_{v,r}$ is impossible—has probability zero—and the decrement term thus adds nothing to our sum. We can therefore simplify further:

(70)
$$\begin{aligned}
&\textstyle\sum_\delta Pr(\Delta(B_t(\top_r),U_t)=\delta|b_v,b_r)\cdot\delta\\
=\;&(f_v+F_v)\times(b_r\times\text{Prob. of correct output with }\top_r\times\gamma(1-b_r)\\
&\qquad+b_r\times\text{Prob. of wrong output with }\top_r\times(-\gamma)b_r\\
&\qquad+(1-b_r)\times\text{Prob. of wrong output with }\bot_r\times\gamma(1-b_r))
\end{aligned}$$

Doing the same expansion for $\sum_\delta Pr(\Delta(B_t(\top_{v,r}),U_t)=\delta|b_v,b_r)\cdot\delta$, we get (71):

(71)
$$\begin{aligned}
&\textstyle\sum_\delta Pr(\Delta(B_t(\top_{v,r}),U_t)=\delta|b_v,b_r)\cdot\delta\\
=\;&f_v\times(b_v\times\text{Prob. of correct output with }\top_{v,r}\times\gamma(1-b_v)\\
&\qquad+b_v\times\text{Prob. of wrong output with }\top_{v,r}\times(-\gamma)b_v\\
&\qquad+(1-b_v)\times\text{Prob. of wrong output with }\bot_{v,r}\times\gamma(1-b_v))
\end{aligned}$$

Finally, let us develop the same sort of formula-in-outline for the remaining summation term, $\sum_{\delta_v\delta_r} Pr(\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)=\delta_v\delta_r|b_v,b_r)$. This will be slightly more complicated than for the previous two formulas, because it requires us working out the probability of each possible value of $\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)$ rather than the just the probabilities of each of the possible changes to individual belief strengths.

First note that we need only consider cases in which v is heard, not cases in which some other verb inflected by r is heard: the value of $\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)$ will always be zero when a verb other than v is heard—regardless of whether it is inflected by r—because the value of $\Delta(B_t(\top_{v,r}),U_t)$ will always be zero in those cases.

Furthermore, although there are in principle sixteen cases of $\Delta(B_t(\top_{v,r}),U_t)\cdot\Delta(B_t(\top_r),U_t)$ to be considered—four cases of $\Delta(B_t(\top_{v,r}),U_t)$, times four of $\Delta(B_t(\top_r),U_t)$)—we need not bother to write out all sixteen. Some we can exclude as having probability zero from the start. We can see this by looking at Yang's inflection-learning model, repeated here as (72).

(72) On processing a past tense verb form X with stem x, to determine whether rule R gives X…

a. Stochastically choose a marking grammoid for x—either allowing or disallowing R's application..

b. Stochastically choose a grammoid telling us whether R does or does not apply at all (or whether it really "exists").

c. Apply/don't apply R to x (falling back on *-d* if we don't apply R):

(i) If the output matches X, reward our beliefs in both x's marking grammoid *and* R's existence grammoid.

(ii) If the output does not match X, penalize our beliefs in both x's marking grammoid *and* R's existence grammoid.

Step 18c in (72) tells us that, after having selected one stem-marking grammoid (either $\top_{v,r}$ or $\bot_{v,r}$) and one rule-existence grammoid (either $\top_r$ or $\bot_r$), the learner will always either reward both grammoids or penalize both grammoids. First, we know, as before, that decrements to the correct grammoids are impossible, since they only come from rewards to the incorrect grammoids, and if either of those is chosen, then neither chosen grammoid will be rewarded.

Furthermore, it cannot happen that both grammoids are penalized: we assume that there are no external factors to be considered so that choosing both correct grammoids would fail to result in a reward for both.

Finally, it is never the case that the learner rewards one selected grammoid, while penalizing the other. This means that certain combinations of belief changes—and thus certain cases of $\Delta(B_t(\top_{v,r}), U_t) \cdot \Delta(B_t(\top_r), U_t)$—have probability zero of occurring, and thus do not need to be considered. We know that there can be no combination of reward with penalty; moreover, since we are considering the changes made to $\top_{v,r}$ and $\top_r$—which include compensations in case $\bot_{v,r}$ or $\bot_r$ is selected—we know more generally that there can be no combination of reward, on the one hand, with penalty *or* compensation, on the other: compensation to one of these correct grammoids happens when the incorrect grammoid is chosen for one of the two linguistic lessons, and is then penalized. No matter what the learner's choice of grammoids for the second of the two linguistic lessons, the learning model makes it impossible for it to be rewarded if the first is being penalized. The correct grammoid for one lesson thus cannot be rewarded when the other is penalized *or* compensated.

All these considerations reduce the number of possible cases of $\Delta(B_t(\top_{v,r}), U_t)\Delta(B_t(\top_r), U_t)$ to four. An outline of a formula for $\sum_{\delta_v\delta_r} Pr(\Delta(B_t(\top_{v,r}), U_t)\Delta(B_t(\top_r), U_t) = \delta_v\delta_r | b_v, b_r)$ is given in (73).

$$\begin{align*}
&\textstyle\sum_{b_v,b_r} Pr(b_v,b_r)\cdot\sum_{\delta_v\delta_r} Pr(\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)=\delta_v\delta_r|b_v,b_r)\cdot\delta_v\delta_r\\
=\;&\text{Prob. of } \top_{v,r} \text{ rew. and } \top_r \text{ rew.} \times (\gamma\cdot(1-b_v))(\gamma\cdot(1-b_r))\\
+\;&\text{Prob. of } \top_{v,r} \text{ pen. and } \top_r \text{ comp.} \times (-\gamma\cdot b_v)(\gamma\cdot(1-b_r))\\
+\;&\text{Prob. of } \top_{v,r} \text{ comp. and } \top_r \text{ pen.} \times (\gamma\cdot(1-b_v))(-\gamma\cdot b_r)\\
+\;&\text{Prob. of } \top_{v,r} \text{ comp. and } \top_r \text{ comp.} \times (\gamma\cdot(1-b_v))(\gamma\cdot(1-b_r))
\end{align*}$$

Which is, as before,

$$\begin{align*}
=\;&\text{Prob. of choosing } \top_{v,r} \text{ and } \top_r \times \text{Prob. of correct output} \times (\gamma\cdot(1-b_v))(\gamma\cdot(1-b_r))\\
+\;&\text{Prob. of choosing } \top_{v,r} \text{ and } \bot_r \times \text{Prob. of incorrect output} \times (-\gamma\cdot b_v)(\gamma\cdot(1-b_r)) \tag{73}\\
+\;&\text{Prob. of choosing } \bot_{v,r} \text{ and } \top_r \times \text{Prob. of incorrect output} \times (\gamma\cdot(1-b_v))(-\gamma\cdot b_r)\\
+\;&\text{Prob. of choosing } \bot_{v,r} \text{ and } \bot_r \times \text{Prob. of incorrect output} \times (\gamma\cdot(1-b_v))(\gamma\cdot(1-b_r))
\end{align*}$$

Which is, in turn,

$$\begin{align*}
=\;& f_v \times (b_v b_r \times \text{Prob. of correct output} \times (\gamma\cdot(1-b_v))(\gamma\cdot(1-b_r))\\
&\quad + b_v(1-b_r) \times \text{Prob. of incorrect output} \times (-\gamma\cdot b_v)(\gamma\cdot(1-b_r))\\
&\quad + (1-b_v)b_r \times \text{Prob. of incorrect output} \times (\gamma\cdot(1-b_v))(-\gamma\cdot b_r)\\
&\quad + (1-b_v)(1-b_r) \times \text{Prob. of incorrect output} \times (\gamma\cdot(1-b_v))(\gamma\cdot(1-b_r)))
\end{align*}$$

This is a formula, in outline, for $\sum_{\delta_v\delta_r} Pr(\Delta(B_t(\top_{v,r}),U_t)\cdot\Delta(B_t(\top_r),U_t)=\delta_v\delta_r|b_v,b_r)$. Now that we have outlines like this for the missing sums in all three expected value formulas, all that remains is to substitute into each the probabilities of the learner's getting a correct/incorrect output in each case.

In (73) this is easy. We assume, as always, that the fact that the choice of $\top_{v,r}$ with $\top_r$ as the target grammar means that it will yield correct output with probability 1; unlike in the previous section, however, we assume that there are no inputs to the learner—at least, none that will lead to any change in beliefs—that are at the same time consistent with the correct grammar *and* some other grammar—a past-tense verb cannot be simultaneously correctly and incorrectly marked. All the missing probabilities in (73), therefore, are 1, giving (74).

$$\begin{aligned}
&\textstyle\sum_{b_v,b_r} Pr(b_v,b_r)\cdot\sum_{\delta_v\delta_r} Pr(\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)=\delta_v\delta_r|b_v,b_r)\cdot\delta_v\delta_r\\
=\ & f_v\times(b_vb_r\times(\gamma(1-b_v))(\gamma(1-b_r))\\
&\quad+b_v(1-b_r)\times(-\gamma b_v)(\gamma(1-b_r))\\
&\quad+(1-b_v)b_r\times(\gamma(1-b_v))(-\gamma b_r)\\
&\quad+(1-b_v)(1-b_r)\times(\gamma(1-b_v))(\gamma(1-b_r)))\\
=\ & \gamma^2 f_v(1-b_rb_v^2-b_r^2b_v+5b_rb_v-2b_v-2b_r)
\end{aligned}\tag{74}$$

The situation is somewhat more complicated for the other sums. The way we have outlined these formulas, the missing piece of information is the probability that *one* of the two grammoids will yield correct/incorrect output when chosen, not that correct/incorrect output will result after having chosen *both* in some way. Supposing that $\top_{v,r}$ is chosen, what is the probability that a correct output will result? $B_t(\top_r)$, of course—the only way to get correct output is to select *both* grammoids correctly. Similarly, supposing that $\top_{v,r}$ is chosen, the probability that *in*correct output will result is $1-B_t(\top_r)$. If $\bot_{v,r}$ is chosen, hope is lost—the probability of incorrect output is 1. (This is what we said before when we pointed out that decrements had probability zero.) Making these changes yields (75) for $\sum_\delta Pr(\Delta(B_t(\top_{v,r}),U_t)=\delta|b_v,b_r)\cdot\delta$:

$$\begin{aligned}
&\textstyle\sum_\delta Pr(\Delta(B_t(\top_{v,r}),U_t)=\delta|b_v,b_r)\cdot\delta\\
=\ & f_v\times(b_v\times b_r\times\gamma(1-b_v)+b_v\times(1-b_r)\times(-\gamma)b_v+(1-b_v)\times\gamma(1-b_v))\\
=\ & \gamma f_v(b_rb_v-2b_v+1)
\end{aligned}\tag{75}$$

Similarly, we have (76) for $\sum_\delta Pr(\Delta(B_t(\top_r),U_t)=\delta|b_v,b_r)\cdot\delta$:

$$\begin{aligned}
&\textstyle\sum_\delta Pr(\Delta(B_t(\top_r),U_t)=\delta|b_v,b_r)\cdot\delta\\
=\ & (f_v+F_v)\times(b_r\times b_v\times\gamma(1-b_r)+b_r\times(1-b_v)\times(-\gamma)b_r+(1-b_r)\times\gamma(1-b_r))\\
=\ & \gamma(f_v+F_v)(b_rb_v-2b_r+1)
\end{aligned}\tag{76}$$

We can now return to our expected values from above. First, substituting (76) into the expression we had for $E[B_t(\top_{v,r})\Delta(B_t(\top_r),U_t)]$, we have:

$$\begin{aligned}
&E[B_t(\top_{v,r})\Delta(B_t(\top_r),U_t)] \\
= \; &\textstyle\sum_{b_v,b_r} Pr(b_v,b_r) \times b_v \times \gamma(f_v+F_v)(b_r b_v - 2b_r + 1)
\end{aligned}$$

which is, by definition of $E[\cdot]$,

$$\begin{aligned}
= \; &E[B_t(\top_{v,r}) \times \gamma(f_v+F_v)(B_t(\top_r)B_t(\top_{v,r}) - 2B_t(\top_r) + 1)] \qquad (77) \\
= \; &E[\gamma(f_v+F_v)(B_t(\top_r)B_t(\top_{v,r})^2 - 2B_t(\top_r)B_t(\top_{v,r}) + B_t(\top_{v,r})]
\end{aligned}$$

which is, by linearity of $E[\cdot]$,

$$= \; \gamma(f_v+F_v)(E[B_t(\top_r)B_t(\top_{v,r})^2] - 2E[B_t(\top_r)B_t(\top_{v,r})] + E[B_t(\top_{v,r})])$$

Similarly, for $E[B_t(\top_r)\cdot\Delta(B_t(\top_{v,r}),U_t)]$, we have:

$$\begin{aligned}
&E[B_t(\top_r)\Delta(B_t(\top_{v,r}),U_t)] \\
= \; &\gamma f_v(E[B_t(\top_r)^2 B_t(\top_{v,r})] - 2E[B_t(\top_r)B_t(\top_{v,r})] + E[B_t(\top_r)])
\end{aligned} \qquad (78)$$

Finally, for $E[\Delta(B_t(\top_{v,r}),U_t)\cdot\Delta(B_t(\top_r),U_t)]$, we have:

$$\begin{aligned}
&E[\Delta(B_t(\top_{v,r}),U_t)\Delta(B_t(\top_r),U_t)] \\
= \; &\gamma^2 f_v(5E[B_t(\top_r)B_t(\top_{v,r})] - 2E[B_t(\top_{v,r})] - 2E[B_t(\top_r)] + 1 \\
&\quad - E[B_t(\top_r)B_t(\top_{v,r})^2] - E[B_t(\top_r)^2 B_t(\top_{v,r})])
\end{aligned} \qquad (79)$$

Putting all of this back together into our formula for $E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)]$, we get:

$$
\begin{aligned}
& E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)] \\
= \;& E[B_t(\top_{v,r})B_t(\top_r)] \\
+ \;& \gamma(f_v+F_v)(E[B_t(\top_r)B_t(\top_{v,r})^2]-2E[B_t(\top_r)B_t(\top_{v,r})]+E[B_t(\top_{v,r})]) \\
+ \;& \gamma f_v(E[B_t(\top_r)^2B_t(\top_{v,r})]-2E[B_t(\top_r)B_t(\top_{v,r})]+E[B_t(\top_r)]) \\
+ \;& \gamma^2 f_v(5E[B_t(\top_r)B_t(\top_{v,r})]-2E[B_t(\top_{v,r})]-2E[B_t(\top_r)]+1 \\
& \qquad -E[B_t(\top_r)B_t(\top_{v,r})^2]-E[B_t(\top_r)^2B_t(\top_{v,r})])
\end{aligned}
$$

(80) which is, simplifying slightly,

$$
\begin{aligned}
= \;& E[B_t(\top_{v,r})B_t(\top_r)] \\
+ \;& \gamma(f_v(\gamma(5E[B_t(\top_r)B_t(\top_{v,r})]-2(E[B_t(\top_{v,r})]+E[B_t(\top_r)])+1 \\
& \qquad -E[B_t(\top_r)B_t(\top_{v,r})^2]-E[B_t(\top_r)^2B_t(\top_{v,r})]) \\
& \quad +E[B_t(\top_{v,r})]+E[B_t(\top_r)]-4E[B_t(\top_r)B_t(\top_{v,r})] \\
& \quad +E[B_t(\top_r)^2B_t(\top_{v,r})]+E[B_t(\top_r)B_t(\top_{v,r})^2)) \\
& +F_v(E[B_t(\top_{v,r})]-2E[B_t(\top_r)B_t(\top_{v,r})]+E[B_t(\top_r)B_t(\top_{v,r})^2]))
\end{aligned}
$$

We have thus derived a formula, (80), which tells us $E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)]$ in terms of constants—just what we wanted—but there is a problem. Recall that we had a similar formula in the previous section. It gave us $E[B_{t+1}(G_i)]$ in terms of constants, including $E[B_t(G_i)]$. We said that this was a *recurrence relation* which had a *closed form solution*—a formula that would tell us $E[B_{t+1}(G_i)]$ for arbitrary t without our having to work out $E[B_t(G_i)]$—which we found.

The formula in (80) is not the same. It tells us $E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)]$ in terms of several constant terms—but not just $E[B_t(\top_{v,r})B_t(\top_r)]$ (the same quantity we are trying to evaluate, but evaluated at the previous time step). Instead, it also makes reference to other previous-time-step expected values which we do not know how to calculate: $E[B_t(\top_r)^2B_t(\top_{v,r})]$, $E[B_t(\top_r)B_t(\top_{v,r})^2]$, $E[B_t(\top_r)]$, and $E[B_t(\top_{v,r})]$.

If we could get *all* of the unknown expected values in the above formula (that is to say, $E[B_{t+1}(\top_r)B_{t+1}(\top_{v,r})]$, $E[B_{t+1}(\top_r)^2B_{t+1}(\top_{v,r})]$, $E[B_{t+1}(\top_r)B_{t+1}(\top_{v,r})^2]$, $E[B_{t+1}(\top_r)]$, and $E[B_{t+1}(\top_{v,r})]$) in terms of *each other*, we might be able to proceed by solving a *system of recurrence relations*. This is not easy in general, but, in this case, it appears to me that even this would be impossible—although I cannot prove it—because of the presence of the $E[B_t(\top_r)^2B_t(\top_{v,r})]$ term: as the reader can verify, if we were to find a formula for $E[B_t(\top_r)^2B_t(\top_{v,r})]$ in terms of constants, including expected values of beliefs at $t-1$, we would find that it contained a $E[B_{t-1}(\top_r)^3 B_{t-1}(\top_{v,r})]$ term; this, in turn, expanded in terms of expected beliefs at time $t-2$, would contain a term with $B_{t-1}(\top_r)$ raised to a higher power; and so on. This prevents us from stating $E[B_t(\top_r)^2 B_t(\top_{v,r})]$ in terms of a finite number of other expected values defined by recurrence relations, as we would need to have a system of recurrence relations we could solve.

Here we begin to see why the promise of a learning model which is precise enough to give predictions analytically is probably not realistic: even a learning system which is, by itself, very simple, like L_{RP}, fails to give clean results when put into the context of a larger model.

We can get a proper system of recurrence relations out of an *approximation* to $E[B_{t+1}(\top_r)B_{t+1}(\top_{v,r})]$, however: if we remove the $E[B_t(\top_r)^2 B_t(\top_{v,r})]$ and $E[B_t(\top_r)B_t(\top_{v,r})^2]$ terms from (80), we get something (hopefully) close to $E[B_{t+1}(\top_r)B_{t+1}(\top_{v,r})]$ in terms only of $E[B_t(\top_r)B_t(\top_{v,r})]$, $E[B_t(\top_r)]$, and $E[B_t(\top_{v,r})]$. $E[B_{t+1}(\top_r)]$ and $E[B_{t+1}(\top_{v,r})]$, in turn, can be shown to be definable in terms only of $E[B_t(\top_r)B_t(\top_{v,r})]$ (for both) and $E[B_t(\top_r)]$ and $E[B_t(\top_{v,r})]$ (respectively), by a procedure exactly parallel to (but much simpler than) the one we used above to find $E[B_{t+1}(\top_r)B_{t+1}(\top_{v,r})]$. I omit the details here and present only the final system of equations.

$$
\begin{aligned}
& E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)] \\
\approx\ & E[B_t(\top_{v,r})B_t(\top_r)] \\
+\ & \gamma(f_v(\gamma(5E[B_t(\top_r)B_t(\top_{v,r})] - 2(E[B_t(\top_{v,r})] + E[B_t(\top_r)]) + 1) \\
& \qquad + E[B_t(\top_{v,r})] + E[B_t(\top_r)] - 4E[B_t(\top_r)B_t(\top_{v,r})]) \\
& \quad + F_v(E[B_t(\top_{v,r})] - 2E[B_t(\top_r)B_t(\top_{v,r})])) \\
\\
& E[B_{t+1}(\top_{v,r})] \\
=\ & E[B_t(\top_{v,r})] + \gamma f_v(E[B_t(\top_r)B_t(\top_{v,r})] - 2E[B_t(\top_{v,r})] + 1) \\
\\
& E[B_{t+1}(\top_r)] \\
=\ & E[B_t(\top_r)] + \gamma(f_v + F_v)(E[B_t(\top_r)B_t(\top_{v,r})] - 2E[B_t(\top_r)] + 1)
\end{aligned}
\tag{81}
$$

In (81), we have a system of recurrence relations—statements of three beliefs at time $t+1$, each defined in terms of itself, and the other two beliefs, at time t. This system, like any other system of equations, can in principle be solved, so that we would have a statement $E[B_{t+1}(\top_{v,r})B_{t+1}(\top_r)]$ not making reference to any of the previous values of these beliefs (except for the beliefs at time $t = 0$). This would tell us, to an approximation, the relation between frequency and beliefs here.

In fact, although we have simplified greatly, we *still* cannot get a reasonable closed-form solution to this system: it appears to be possible, but the best result reported by a computer algebra program is far too long and complex to be useful.

Nevertheless, we set out to get *some* testable, quantitative hypothesis of the free-rider effect. To do this, I will take a further approximation to the system: I will approximate $E[B_{t+1}(\top_{v,r})]$ and $E[B_{t+1}(\top_r)]$ by removing the $E[B_t(\top_r)B_t(\top_{v,r})]$ terms in the expression above, by removing the $E[B_t(\top_r)]$ terms, thus obtaining approximations to these quantities in terms of their respective previous values.

$$
\begin{aligned}
E[B_{t+1}(\top_{v,r})] &\approx \gamma f_v(1-2E[B_t(\top_{v,r})]) \\
E[B_{t+1}(\top_r)] &\approx \gamma(f_v+F_v)(1-2E[B_t(\top_r)])
\end{aligned} \tag{82}
$$

This simpler system has a much simpler solution (again, derived by computer), but it is still too large to be presented here, and it is not in a form that is useful for us for doing statistical tests. It is, however, some sort of polynomial in f_v and F_v. We will keep this in mind.

Now recall that we are ultimately trying to match the data to a statistical model. One very simple model is a *linear regression*.

$$
Y = \beta_0+\beta_1X_1+\beta_2X_2+\varepsilon \tag{83}
$$

A linear regression assumes that the observed values of the dependent variable result from summing together one or more independent variables, each multiplied by a constant, along with some other constant β_0, plus some random error ε. To apply this model here, we need to make some assumptions to get our (vague) prediction in the appropriate form. The data are the measurements of $B_t(\top_{v,r})B_t(\top_r)$, of f_v, and of F_v; we would like our model to predict $B_t(\top_{v,r})B_t(\top_r)$ based on the frequency data, so we would like the two frequency variables to be the independent variables, or to combine in some way to give us a single independent variable. We are free to do whatever we like to a set of data points before making it one of the variables in the regression; perhaps there is a way of making a linear model look like some polynomial in f_v and F_v by applying some function(s) to the data:

$$
G(Y) = \beta_0+\beta_1G_1(X_1,X_2)+\beta_2G_2(X_1,X_2)+\varepsilon \tag{84}
$$

Suppose we take the logarithm of each of the variables. Then we get (85).

$$\begin{aligned} \log(B_t(\top_{v,r})B_t(\top_r)) &= \beta_0 + \beta_1 \log f_v + \beta_2 \log F_v \\ &= \beta_0 + \log f_v^{\beta_1} + \log F_v^{\beta_2} \\ B_t(\top_{v,r})B_t(\top_r) &= e^{\beta_0} f_v^{\beta_1} F_v^{\beta_2} \end{aligned} \tag{85}$$

If we do this, we predict that beliefs are related to the two important frequencies by taking each, raising them to some power, then multiplying them together and multiplying them by a constant. If a term containing both f_v and F_v is dominant in the polynomial representing our prediction, then this is a good approximation. There is no good way to tell if this is true—but there are no ways of approximating a solution to our system that are obviously better than this as far as I can tell.

We thus have—or seem to have—some idea what the relation among the variables ought to be under Yang's model. Perhaps more importantly, we have seen the apparent unavailability of a closed-form solution to the real prediction of the model, and the enormous complexity of deriving any prediction at all, even in this relatively simple case; this shows that, although Variational Learning assuming the L_{RP} system can *in principle* give us predictions analytically, but this is really not practical in realistic cases.

Appendix B

Data from CHILDES/Marcus et al. 1992

Each child's relative parental frequency (R), and correct usage rate (versus both kinds of overregularization discussed above: C) on each verb is presented; correct usage rates are not presented for verbs which appeared less than ten times in that child's speech, but these verbs were left in the analysis on the grounds that they still contributed to class frequency. Parental frequency data for *get–got* were extracted for a subset of the children in the first study (Adam, Eve, and Abe, but not Sarah, April, Naomi, Nat, Nathaniel, or Peter).

Adam	Brown 1973
Eve	Brown 1973
Sarah	Brown 1973
Abe	Kuczaj 1977
April	Higginson 1985
Naomi	Sachs 1983
Nat	Bohannon and Marquis 1977
Nathaniel	MacWhinney 2000
Peter	Bloom *et al.* 1974

Verb	Adam		Eve		Sarah		Abe		April		Naomi		Nat		Nathaniel		Peter	
	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C
beat	0.0000%	–	0.0000%	–	0.0000%	–	0.0181%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
bite	0.0262%	100%	0.0338%	–	0.0043%	100%	0.0090%	–	0.3030%	–	0.0163%	–	0.0000%	–	0.0095%	–	0.0096%	–
bleed	0.0000%	–	0.0000%	–	0.0021%	–	0.0045%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
blow	0.0112%	–	0.0000%	–	0.0043%	–	0.0000%	31%	0.0000%	–	0.0000%	–	0.0000%	–	0.0569%	–	0.0000%	–
break	0.1611%	100%	0.0811%	100%	0.1327%	100%	0.0722%	70%	0.1212%	–	0.0898%	100%	0.1081%	–	0.0190%	71%	0.0736%	93%
bring	0.0562%	100%	0.0135%	–	0.0985%	–	0.0361%	–	0.0606%	–	0.0163%	–	0.0270%	–	0.0759%	–	0.0863%	–
buy	0.0150%	100%	0.0338%	–	0.1370%	86%	0.0226%	58%	0.1818%	–	0.0082%	–	0.0270%	–	0.1233%	–	0.0064%	–
catch	0.0599%	97%	0.0203%	–	0.0364%	94%	0.0632%	79%	0.0606%	–	0.0082%	–	0.0000%	–	0.0474%	–	0.0416%	–
choose	0.0000%	–	0.0000%	–	0.0000%	–	0.0045%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
come	0.1423%	99%	0.1148%	85%	0.1177%	79%	0.2032%	26%	0.0303%	–	0.0898%	–	0.1622%	–	0.2798%	100%	0.1407%	100%
cut	0.0000%	92%	0.0338%	–	0.0064%	–	0.0406%	56%	0.0000%	–	0.0000%	–	0.0000%	–	0.0285%	–	0.0096%	–
dig	0.0000%	–	0.0068%	–	0.0021%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
draw	0.0150%	–	0.0338%	–	0.0150%	–	0.0316%	–	0.0000%	–	0.0326%	–	0.0000%	–	0.0000%	–	0.0224%	–
drink	0.0075%	–	0.0270%	–	0.0064%	–	0.0406%	50%	0.0303%	–	0.0082%	–	0.0000%	–	0.0047%	–	0.0032%	–
drive	0.0000%	–	0.0000%	–	0.0043%	–	0.0090%	–	0.0303%	–	0.0000%	–	0.0000%	–	0.0617%	–	0.0064%	–
eat	0.0375%	100%	0.0270%	–	0.0428%	100%	0.2528%	80%	0.1212%	–	0.0653%	–	0.0000%	–	0.0711%	–	0.0192%	–
fall	0.2023%	98%	0.1148%	20%	0.0535%	91%	0.1580%	56%	0.0909%	–	0.2611%	96%	0.1892%	100%	0.1660%	100%	0.2494%	90%
feed	0.0037%	–	0.0000%	–	0.0064%	–	0.0045%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0190%	–	0.0000%	–
feel	0.0000%	–	0.0000%	–	0.0021%	–	0.0361%	31%	0.0303%	–	0.0000%	–	0.0270%	–	0.0142%	–	0.0064%	–
fight	0.0000%	–	0.0000%	–	0.0043%	–	0.0135%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–

Verb	Adam		Eve		Sarah		Abe		April		Naomi		Nat		Nathaniel		Peter	
	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C
find	0.0674%	100%	0.0203%	–	0.0535%	100%	0.2302%	98%	0.1515%	–	0.0571%	80%	0.0000%	–	0.0664%	100%	0.1407%	100%
fly	0.0037%	–	0.0068%	–	0.0107%	–	0.0226%	–	0.0303%	–	0.0163%	–	0.0000%	–	0.0095%	–	0.0000%	–
forget	0.0449%	100%	0.1621%	100%	0.1113%	100%	0.1129%	100%	0.0909%	–	0.0163%	–	0.0541%	–	0.0474%	–	0.0480%	–
freeze	0.0000%	–	0.0000%	–	0.0021%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
get	0.1461%	100%	0.1418%	100%	–	–	0.3703%	78%	–	100%	–	99%	–	100%	–	100%	–	100%
give	0.0974%	100%	0.0270%	–	0.2076%	100%	0.0677%	–	0.2121%	–	0.0898%	–	0.2703%	–	0.1328%	–	0.0736%	–
grind	0.0000%	–	0.0203%	–	0.0300%	–	0.0316%	–	0.0303%	–	0.0245%	–	0.0541%	–	0.1518%	–	0.0192%	–
grow	0.0000%	–	0.0068%	–	0.0086%	–	0.0181%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0142%	–	0.0000%	–
hang	0.0037%	–	0.0000%	–	0.0000%	–	0.0316%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0095%	–	0.0032%	–
hear	0.0899%	100%	0.0338%	–	0.0471%	67%	0.1038%	33%	0.0303%	–	0.0163%	–	0.0000%	–	0.0047%	–	0.0544%	–
hide	0.0000%	–	0.0000%	–	0.0064%	–	0.0090%	–	0.0000%	–	0.0000%	–	0.0270%	–	0.0000%	–	0.0032%	–
hit	0.0974%	100%	0.0270%	–	0.0428%	100%	0.0587%	67%	0.0000%	–	0.0245%	–	0.0000%	–	0.0237%	–	0.0128%	–
hold	0.0000%	–	0.0000%	–	0.0043%	–	0.0135%	–	0.0000%	–	0.0408%	–	0.0000%	–	0.0047%	–	0.0032%	–
hurt	0.0524%	100%	0.0203%	–	0.0043%	90%	0.0361%	89%	0.0606%	–	0.0490%	–	0.0000%	–	0.0285%	–	0.0320%	–
keep	0.0037%	–	0.0068%	–	0.0235%	–	0.0135%	–	0.0000%	–	0.0082%	–	0.0000%	–	0.0190%	–	0.0128%	–
know	0.0262%	–	0.0000%	–	0.0621%	–	0.0767%	65%	0.0303%	–	0.0326%	–	0.0541%	–	0.0095%	–	0.0192%	–
leave	0.0599%	100%	0.0338%	–	0.1113%	–	0.1445%	87%	0.0000%	–	0.0979%	–	0.1081%	–	0.1138%	–	0.1311%	100%
let	0.0000%	–	0.0000%	–	0.0000%	–	0.0045%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0047%	–	0.0000%	–
light	0.0000%	–	0.0000%	–	0.0021%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0095%	–	0.0000%	–
lose	0.1498%	100%	0.0135%	–	0.1113%	95%	0.0587%	–	0.0909%	–	0.0326%	–	0.0811%	–	0.0617%	–	0.0927%	100%

Verb	Adam		Eve		Sarah		Abe		April		Naomi		Nat		Nathaniel		Peter	
	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C
make	0.3034%	95%	0.0878%	–	0.1434%	87%	0.4379%	82%	0.3030%	–	0.2040%	83%	0.1081%	–	0.1612%	100%	0.1087%	100%
mean	0.0150%	–	0.0000%	–	0.0193%	–	0.0271%	–	0.0303%	–	0.0326%	–	0.0000%	–	0.0190%	–	0.0160%	–
meet	0.0150%	–	0.0000%	–	0.0150%	–	0.0271%	–	0.0303%	–	0.0082%	–	0.0270%	–	0.0617%	–	0.0032%	–
put	0.3184%	100%	0.1283%	100%	0.1263%	98%	0.0993%	88%	0.2121%	–	0.1632%	–	0.0541%	–	0.1423%	–	0.1503%	–
ride	0.0375%	–	0.0000%	–	0.0043%	–	0.0000%	–	0.0303%	–	0.0082%	–	0.0000%	–	0.0000%	–	0.0128%	–
ring	0.0037%	–	0.0000%	–	0.0021%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0047%	–	0.0032%	–
rise	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0047%	–	0.0000%	–
run	0.0524%	90%	0.0068%	–	0.0107%	–	0.0271%	40%	0.0000%	–	0.0245%	–	0.0541%	–	0.0996%	–	0.0192%	–
say	0.4682%	100%	0.1418%	100%	0.4024%	100%	0.5056%	99%	0.1515%	100%	0.4814%	–	0.1081%	–	0.3130%	100%	0.2782%	100%
see	0.2734%	100%	0.0338%	–	0.1520%	100%	0.4560%	96%	0.2727%	–	0.0408%	–	0.4595%	–	0.1707%	–	0.0831%	100%
send	0.0000%	–	0.0135%	–	0.0107%	–	0.0271%	–	0.0909%	–	0.0000%	–	0.0000%	–	0.0332%	–	0.0064%	–
shake	0.0000%	–	0.0068%	–	0.0021%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0095%	–	0.0000%	–
shoot	0.0262%	100%	0.0000%	–	0.0150%	–	0.0361%	77%	0.0303%	–	0.0245%	–	0.0000%	–	0.0142%	–	0.0000%	–
shrink	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
shut	0.0037%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
sing	0.0037%	–	0.0000%	–	0.0086%	–	0.0090%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0237%	–	0.0000%	–
sit	0.0300%	–	0.0000%	–	0.0214%	–	0.0045%	–	0.0303%	–	0.0163%	–	0.0270%	–	0.0617%	–	0.0128%	–
sleep	0.0037%	–	0.0135%	–	0.0150%	–	0.0135%	–	0.0000%	–	0.0245%	–	0.0000%	–	0.0095%	–	0.0000%	–
slide	0.0000%	–	0.0000%	–	0.0150%	–	0.0000%	–	0.0000%	–	0.0163%	–	0.0000%	–	0.0047%	–	0.0000%	–
spend	0.0037%	–	0.0000%	–	0.0086%	–	0.0135%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0142%	–	0.0000%	–

Verb	Adam		Eve		Sarah		Abe		April		Naomi		Nat		Nathaniel		Peter	
	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C	R	C
spin	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
spit	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
stand	0.0000%	–	0.0000%	–	0.0043%	–	0.0045%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0095%	–	0.0000%	–
steal	0.0000%	–	0.0000%	–	0.0021%	–	0.0226%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0047%	–	0.0000%	–
stick	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
string	0.0037%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
sweep	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0082%	–	0.0000%	–	0.0000%	–	0.0000%	–
swim	0.0000%	–	0.0000%	–	0.0043%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
swing	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–
take	0.1648%	97%	0.0675%	–	0.1648%	95%	0.1580%	70%	0.0909%	–	0.0816%	–	0.1622%	–	0.1186%	100%	0.1407%	87%
teach	0.0000%	–	0.0000%	–	0.0364%	–	0.0181%	–	0.0000%	–	0.0000%	–	0.0270%	–	0.0000%	–	0.0064%	–
tear	0.0075%	–	0.0203%	–	0.0000%	–	0.0045%	–	0.0000%	–	0.0082%	–	0.0000%	–	0.0000%	–	0.0224%	–
tell	0.0861%	100%	0.0068%	–	0.2012%	96%	0.2167%	88%	0.0606%	–	0.0979%	–	0.0270%	–	0.0474%	–	0.0448%	–
think	0.2659%	100%	0.1081%	–	0.3018%	100%	0.6050%	81%	0.1212%	–	0.2040%	–	0.1081%	–	0.1280%	–	0.3998%	–
throw	0.0300%	–	0.0135%	–	0.0278%	30%	0.0316%	29%	0.0000%	–	0.0571%	–	0.0000%	–	0.0190%	–	0.0224%	–
wake	0.0112%	–	0.0068%	–	0.0064%	–	0.0135%	–	0.0000%	–	0.0326%	–	0.0000%	–	0.0285%	–	0.0128%	–
wear	0.0000%	–	0.0000%	–	0.0128%	–	0.0135%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0237%	–	0.0000%	–
win	0.0075%	–	0.0000%	–	0.0364%	–	0.0722%	64%	0.0000%	–	0.0000%	–	0.0270%	–	0.0000%	–	0.0416%	–
wind	0.0075%	–	0.0000%	–	0.0021%	–	0.0090%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0000%	–	0.0032%	–
write	0.0337%	100%	0.0135%	–	0.0321%	–	0.0135%	–	0.0000%	–	0.0082%	–	0.0000%	–	0.0095%	–	0.0224%	–
Total Input	26698		14805		46721		22150		3300		12256		3700		21086		31269	

Appendix C

Classifications of English Strong Verbs

BS	*Bybee and Slobin 1982*
Y	*Yang 2002 (starred entries were not included by Yang but are consistent with his presentation)*
M1	*Marcus* et al. *1992, rhyming stem and past*
M2a	*Marcus* et al. *1992, shared final stem coda and same change (broad)*
M2b	*Marcus* et al. *1992, shared final stem coda and same change (narrow)*
M3a	*Marcus* et al. *1992, shared final stem consonant and same change (broad)*
M3b	*Marcus* et al. *1992, shared final stem consonant and same change (narrow)*
HM	*Halle and Mohanan 1985 (rules separated by commas)*

Verb	**BS**	**Y**	**M1**	**M2a**	**M2b**	**M3a**	**M3b**	**HM**
beat	I	-ø/No Change	it–it	V–t	V3–t	V–t	V3–t	-ø
bite	V	-ø/Shortening	ajt–ɪt	I–t	I2–t	I–t	I2–t	-*t*
bleed	V	-ø/Shortening	id–ɛd	I–d	I2–d	I–d	I2–d	-*d*
blow	VIII	-ø/*V* → *u*	o–u	IV–ø	IV1–ø	IV–ø	IV1–ø	-ø,133,131
break	VII	-ø/*V* → *o* *	ek–ok	II–k	II2–k	II–k	II2–k	-ø,131
bring	IV	-*t*/Rime → ɑ	iŋ–ɑt	I–ŋ	I3d–ŋ	I–ŋ	I3d–ŋ	-*t*,131,136,125
buy	IV	-*t*/Rime → ɑ	aj–ɑt	I–ø	I3d–ø	I–ø	I3d–ø	-*t*,131,136

Verb	BS	Y	M1	M2a	M2b	M3a	M3b	HM
catch	IV	-*t*/Rime → ɑ	ætʃ–ɑt	I–tʃ	I3d–tʃ	I–tʃ	I3d–tʃ	-*t*,131
choose	V	-ø/Lowering	uz–oz	II–z	II2–z	II–z	II2–z	-ø,136
come	VII	-ø/Umlaut	ʌm–em	V–m	V2–m	V–m	V2–m	-ø,136
cut	I	-ø/No Change	ʌt–ʌt	I–t	I1–t	I–t	I1–t	-ø
dig	VI	-ø/Backing	ɪg–ʌg	III–g	III2–g	III–g	III2–g	-ø,131
draw	VIII	-ø/*V* → *u*	ɑ–u	IV–ø	IV1–ø	IV–ø	IV1–ø	-ø
drink	VI	-ø/Lowering	ɪŋk–æŋk	III–ŋk	III1–ŋk	III–k	III1–k	-ø,136
drive	VII	-ø/Backing	ajv–ov	IV–v	IV4–v	IV–v	IV4–v	-ø,131,136
eat	V	-ø/Lowering	it–et	V–t	V3–t	V–t	V3–t	-ø,136
fall	VII	-ø/Umlaut	al–ɛl	V–l	V2–l	V–l	V2–l	-ø
feed	V	-ø/Shortening	id–ɛd	I–d	I2–d	I–d	I2–d	-*d*
feel	III	-*t*/Shortening	il–el	I–l	I3c–l	I–l	I3c–l	-*t*
fight	IV	-*t*/Rime → ɑ	ajt–t	I–t	I3d–t	I–t	I3d–t	-*t*,136,131
find	V	-ø/Backing	ajnd–awnd	IV–nd	IV3–nd	IV–d	IV3–d	-ø,131
fly	VIII	-ø/*V* → *u*	aj–u	IV–ø	IV1–ø	IV–ø	IV1–ø	-ø,131,133
forget	V	-ø/Backing	ɛt–ɑt	II–t	II1–t	II–t	II1–t	-ø,131
freeze	VII	-ø/Backing	iz–oz	II–z	II1–z	II–z	II1–z	-ø,136
get	V	-ø/Backing	ɛt–ɑt	II–t	II1–t	II–t	II1–t	-ø,131
give	VII	-ø/*V* → *e* *	ɪv–ev	V–v	V3–v	V–v	V3–v	-ø,133,136
grind	V	-ø/Backing	ajnd–awnd	IV–nd	IV3–nd	IV–d	IV3–d	-ø,131
grow	VIII	-ø/*V* → *u*	o–u	IV–ø	IV1–ø	IV–ø	IV1–ø	-ø,133,131
hang	VI	-ø/*V* → ʌ	æŋ–ʌŋ	III–ŋ	III3–ŋ	III–ŋ	III3–ŋ	-ø,136
hear	III	-*d*/Shortening	ir–ʌrd	I–r	I4a–r	I–r	I4a–r	-*d*
hide	V	-ø/Shortening	ajd–ɪd	I–d	I2–d	I–d	I2–d	-*d*
hit	I	-ø/No Change	ɪt–ɪt	I–t	I1–t	I–t	I1–t	-ø
hold	V	-ø/Umlaut	old–ɛld	IV–ld	IV2–ld	IV–d	IV2–d	-ø,131
hurt	I	-ø/No Change	ʌrt–ʌrt	I–rt	I1–rt	I–t	I1–t	-ø
keep	III	-*t*/Shortening	ip–ɛpt	I–p	I3c–p	I–p	I3c–p	-*t*
know	VIII	-ø/*V* → *u*	o–u	IV–ø	IV1–ø	IV–ø	IV1–ø	-ø,133,131
leave	III	-*t*/Shortening	iv–ɛft	I–v	I3c–v	I–v	I3c–v	-*t*
let	I	-ø/No Change	ɛt–ɛt	I–t	I1–t	I–t	I1–t	-ø
light	V	-ø/Shortening	ajt–ɪt	I–t	I2–t	I–t	I2–t	-*t*

Verb	**BS**	**Y**	**M1**	**M2a**	**M2b**	**M3a**	**M3b**	**HM**
lose	III	-*t*/Shortening	uz–ɑst	I–z	I3c–z	I–z	I3c–z	-*t*
make	III	-*d*/Deletion	ek–ed	I–k	I4b–k	I–k	I4b–k	-*d*,125
mean	III	-*t*/Shortening	in–ɛnt	I–n	I3c–n	I–n	I3c–n	-*t*
meet	V	-ø/Shortening	it–ɛt	I–t	I2–t	I–t	I2–t	-*t*
put	I	-ø/No Change	ʊt–ʊt	I–t	I1–t	I–t	I1–t	-ø
ride	V	-ø/Backing	ajd–od	IV–d	IV4a–d	IV–d	IV4a–d	-ø,133,131
ring	VI	-ø/Lowering	ɪŋ–æŋ	III–ŋ	III1–ŋ	III–ŋ	III1–ŋ	-ø,136
rise	VII	-ø/Backing	ajz–oz	IV–z	IV4a–z	IV–z	IV4a–z	-ø,131,133
run	VII	-ø/*V* → *æ* *	ʌn–æn	III–n	III3–n	III–n	III3–n	-ø,136
say	III	-*d*/Shortening	e–ɛd	I–ø	I4a–ø	I–ø	I4a–ø	-*d*,133
see	VIII	-ø/Rime → ɑ *	i–ɑ	V–ø	V3–ø	V–ø	V3–ø	-ø,133,131
send	II	-*t*/Deletion	ɛnd–ɛnt	I–nd	I3b–d	I–nd	I3b–nd	-*t*
shake	VII	-ø/Backing	ek–ʊk	IV–k	IV2–k	IV–k	IV2–k	-ø,131
shoot	V	-ø/Shortening	ut–at	I–t	I2–t	I–t	I2–t	-*t*
shrink	VI	-ø/Lowering	ɪŋk–æŋk	III–ŋk	III1–ŋk	III–k	III1–k	-ø,136
shut	I	-ø/No Change	ʌt–ʌt	I–t	I1–t	I–t	I1–t	-ø
sing	VI	-ø/Lowering	ɪŋ–æŋ	III–ŋ	III1–ŋ	III–ŋ	III1–ŋ	-ø,136
sit	VI	-ø/Lowering	ɪt–æt	V–t	V4–t	V–t	V4–t	-ø,136
sleep	III	-*t*/Shortening	ip–ɛpt	I–p	I3c–p	I–p	I3c–p	-*t*
slide	V	-ø/Shortening	ajd–ɪd	I–d	I2–d	I–d	I2–d	-*d*
spend	II	-*t*/Deletion	ɛnd–ɛnt	I–nd	I3b–nd	I–d	I3b–d	-*t*
spin	VI	-ø/Backing *	ɪn–ʌn	III–n	III2–n	III–n	III2–n	-ø,133
spit	V	-ø/Lowering	ɪt–æt	V–t	V4–t	V–t	V4–t	-ø,136
stand	V	-*d*/*stood* rule	ænd–ʊd	V–nd	V4b–nd	V–nd	V4b–nd	-ø,*stood*
steal	VII	-ø/Backing	il–ol	II–l	II1–l	II–l	II1–l	-ø,131,136
stick	VI	-ø/Backing	ɪk–ʌk	III–k	III2–k	III–k	III2–k	-ø,131
string	VI	-ø/Backing	ɪŋ–ʌŋ	III–ŋ	III2–ŋ	III–ŋ	III2–ŋ	-ø,131
sweep	III	-*t*/Shortening	ip–ɛpt	I–p	I3c–p	I–p	I3c–p	-*t*
swim	VI	-ø/Lowering	ɪm–æm	III–m	III1–m	III–m	III1–m	-ø,136
swing	VI	-ø/Backing	ɪŋ–ʌŋ	III–ŋ	III2–ŋ	III–ŋ	III2–ŋ	-ø,131
take	VII	-ø/Backing	ek–ʊk	IV–k	IV2–k	IV–k	IV2–k	-ø,131
teach	IV	-*t*/Rime → ɑ	itʃ–ɑt	I–tʃ	I3d–tʃ	I–tʃ	I3d–tʃ	-*t*,125,131

Verb	**BS**	**Y**	**M1**	**M2a**	**M2b**	**M3a**	**M3b**	**HM**
tear	VI	-ø/Backing	ɛr–or	II–r	II1–r	II–r	II1–r	-ø,131
tell	III	-*d*/Backing	ɛl–old	I–l	I4c–l	I–l	I4c–l	-*d*,131
think	IV	-*t*/Rime → ɑ	ɪŋk–ɑt	I–ŋk	I3d–ŋk	I–k	I3d–k	-*t*,125,131
throw	VIII	-ø/V → u	o–u	IV–ø	IV1–ø	IV–ø	IV1–ø	-ø,133,131
wake	VII	-ø/V → o *	ek–ok	II–k	II2–k	II–k	II2–k	-ø,131
wear	VII	-ø/Backing	ɛr–or	II–r	II1–r	II–r	II1–r	-ø,131
win	VI	-ø/Backing	ɪn–ʌn	III–n	III2–n	III–n	III2–n	-ø,131
wind	V	-ø/Backing	ajnd–awnd	IV–nd	IV3–nd	IV–d	IV3–d	-ø,131
write	V	-ø/Backing	ajt–ot	IV–t	IV4a–t	IV–t	IV4a–t	-ø,136,131

Appendix D

The Learner, with Examples

The complete logic of the learner is given in (86). Steps that are exclusive to DL are *italicized.*

(86) On input $\langle s,t \rangle$:

a. Predict the past tense of *s* using a currently favoured set of grammoids:

1. *Rule existence grammoids, which tell us what rules to include in the derivation.*
2. Rule morpholexicality grammoids, which tell us whether those rules are morpholexical.
3. Features on *s* determining whether it should be included in a rule's domain (for morpholexical rules) or excluded (for non-morpholexical rules).

b. Based on the attested output, *t*, and the predicted output, compute error signals that will tell the learner how it should change its grammar:

1. Error signals telling the learner what is missing from its rule system (one for internal changes and one for suffixation). (See (91) and (95).)
2. *Error signals for existence grammoids, which tell the learner the direction in which belief in those grammoids should be adjusted.* (See (112).)
3. Error signals for rule morpholexicality grammoids, which tell the learner the direction in which belief in those grammoids should be adjusted. See (103).
4. Error signals for the rule features on *s*, telling the learner the direction in which belief in the existing grammoids should be adjusted, or whether new ones should be added. (See (108).)

c. Change the grammar using the error signal:

   1. Use the rule error signals to update the set of rules.

      (a) Use the rule error signals to determine whether new rules should be added, or whether an existing rule should be changed, then make the appropriate changes. (See (98).)

      (b) Use the rule error signals to determine whether duplicate rules should be added, or whether a new rule should be added—or neither—then add the necessary rules, giving each an existence grammoid. (See section (3.3.2) and (98).)

   2. Use the existence error signals to adjust belief in the existence of each rule. (See (101).)

   3. Use the rule morpholexicality error signals to adjust belief in the morpholexicality of each rule. (See (101).)

   4. Use the current rule feature error signals to adjust belief in existing rule features, or to add new rule features to *s*. (See (101).)

Full Example

A sample run of ML, taken from a real run.[1]

(87)

1. Input is $\langle$fajnd, FIND, fawnd$\rangle$. Current beliefs and the selection are as in (88).
2. Derivation is as in (89).
3. Calculate errors:
   (a) Use (95) on $\langle$fawnd, fawnd$\rangle$ to extract the suffixation error: it returns the error $\text{ø}/[\text{fawnd}] + \_$.
   (b) Use (91) on $\langle$#fajnd#, #fawnd#$\rangle$ (since [fawnd] is the environment in the suffixation error) to extract the internal change error: it returns aj $\rightarrow$ aw/#f_nd#.
4. Update model: $\text{ø}/[\text{fawnd}] + \_$ can combine with $\text{ø}/[-\text{lat}, -\text{str}] + \_$, but this yields the existing rule and does not alter the set of rules; aj $\rightarrow$ aw/#f_nd# can combine with aj $\rightarrow$ aw/#f_nd#, but this yields the existing rule and does not alter the set of rules; changes to beliefs are as in (90).

[1] This learner was biased towards making internal change rules morpholexical, with an initial belief strength of 0.9, and had initial belief strengths of 0.15 for inclusions and 0.3 for exclusions; it had run through one epoch of Mixed 5 when this test was taken. Most internal changes are omitted here to save space.

The steps in (88) show the learner's selections for each of the stochastic grammoids, for each rule. (Recall that in ML, unlike in DL, all the rules are selected on every derivation; there are no existence lessons.) On the fourth line, for example, the learner needs to guess whether the *-d* suffix is morpholexical or not; this is easy, because its belief in morpholexicality for this rule is zero, so it chooses *false*. It must also decide whether the current stem, *find*, has a marked exception feature for this rule. It has a substantial belief strength here (0.5), but in this case chooses *false*. There is no rule feature to be chosen, because the rule is not morpholexical.

(88)

	Rule	*Morpholexical*		FIND *marked with exclusion*		FIND *marked with inclusion*	
(1)	ɪ → æ/[+ant, +cont, +cor, −dor, −lab, −lat, −nas]_ŋk#	B(True) = 0.82	*Selected **True***	*None posited*	—	B(True) = 0.18	*Selected **False***
(2)	ø/[−lat, −str] + _	B(True) = 0.95	*Selected **True***	*None posited*	—	*None posited*	—
(3)	[t]/[−lat, −nas, −son, −voice] + _	B(True) = 0	*Selected **False***	*None posited*	—	*None posited*	—
(5)	[d]/[+voice] + _	B(True) = 0	*Selected **False***	B(True) = 0.49	*Selected **True***	*None posited*	—
(8)	[ɪd]/[+ant, −cont, +cor, −dor, −lab, −lat, −nas, −son, −str] + _	B(True) = 0	*Selected **False***	B(True) = 0.5	*Selected **False***	*None posited*	—
(12)	aj → aw/#f_nd#	B(True) = 0.91	*Selected **True***	*None posited*	—	B(True) = 0.11	*Selected **False***

The steps in (89) show the learner's derivation. The column at the far right shows the input to/output of each rule. Note that the rules are ordered in decreasing order of specificity, even in the conjunctive (internal change) component, to avoid the issue of learning extrinsic ordering. Thus the first suffix, -ø, does not apply because the stem is not marked, and so we say that suffix *skips* the form; the [ɪd] suffix does apply (and thus neither of the following suffixes apply), leading to a suffixation error.

(89)

	Skips?	*Applies?*		UR: [fajnd] ([-5])
(12)	*Yes*	*No*	aj → aw/#f_nd# (*)	[#fajnd#]
(1)	*Yes*	*No*	ɪ → æ/[+ant, +cont, +cor, −dor, −lab, −lat, −nas]_ŋk# (*)	[#fajnd#]
				[fajnd]
(2)	*Yes*	*No*	ø/[−lat, −str] + _ (*)	[fajnd]
(8)	*No*	*Yes*	[ɪd]/[+ant, −cont, +cor, −dor, −lab, −lat, −nas, −son, −str] + _	[fajnd] + [ɪd]
(3)	*No*	*No*	[t]/[−lat, −nas, −son, −voice] + _	—
(5)	*No*	*No*	[d]/[+voice] + _	—
				[fajndɪd]

In (90) are the error signals sent back to the learner for each stochastic linguistic lesson. For example, there is no evidence about the [d] suffix's morpholexicality, because it was never considered, and therefore has no status as having applied or skipped. On the other hand, the ɪd suffix did apply, and there was a suffixation error, so we encourage the rule to be morpholexical, and encourage an exception feature on *find* in case we find ourselves in the same situation again. (This learner is currently on the wrong track, of course, because it is taking the ɪd suffix to be idiosyncratic, but it is likely to recover.)

(90)

	Rule		*Morpholexical*	FIND *marked exception*	FIND *marked as included*
(1)	ɪ → æ/[+ant, +cont, +cor, −dor, −lab, −lat, −nas]_ŋk#	*Skips*	*No evidence*	—	—
(2)	ø/[−lat, −str]+_	*Skips*	[+ML] *selected, but bad*	—	*Evidence for* [+2]
(3)	[t]/[−lat, −nas, −son, −voice]+_	—	*No evidence*	—	—
(5)	[d]/[+voice]+_	—	*No evidence*	—	—
(8)	[ɪd]/[+ant, −cont, +cor, −dor, −lab, −lat, −nas, −son, −str]+_	*Applies*	[−ML] *selected, but bad*	*Evidence for* [−8]	—
(12)	aj → aw/#f_nd#	*Skips*	[+ML] *selected, but bad*	—	*Evidence for* [+12]

Internal Change Error

The procedure in (91) calculates internal change error.

```
(91)  On input $\langle u,v \rangle$:
    1. Let $m$ be the number of segments in $u$ and $n$ the number of segments in $v$. Then $u = u_1 \ldots u_m$ and $v = v_1 \ldots v_n$.
    2. Let $c \leftarrow 0$.
    3. For $i$ in $1,\ldots,\max(m,n)$:
        (a) If $i \leq m$, $U_i \leftarrow u_i$; otherwise, let $U_i \leftarrow \perp$.
        (b) If $i \leq n$, $V_i \leftarrow v_i$; otherwise, let $V_i \leftarrow \perp$.
        (c) If $U_i \neq V_i$, $c \leftarrow i$. Break out of the loop.
    4. If $c = 0$, return $\varnothing \rightarrow \varnothing/\_$.
    5. Let $d_u \leftarrow m$; let $d_v \leftarrow n$.
    6. For $i$ in $c,\ldots m$:
        (a) For $j$ in $c,\ldots,n$:
            i. If $u_i = v_j$, let $d_u \leftarrow i-1$, and let $d_v \leftarrow j-1$. Break out of both loops.
    7. Let $A \leftarrow u_c \ldots u_{d_u}$; let $B \leftarrow v_c \ldots v_{d_v}$; let $C \leftarrow u_1 \ldots u_{c-1}$; let $D \leftarrow u_{d_{u+1}} \ldots u_m$; return $A \rightarrow B/C\_D$.
```

Application of (91) to ⟨#rɪŋ#, #ræŋ#⟩ (*ring/rang*).

(92)

i	j	c	d_u	d_v	u	v	Comment
1		0			$\underline{\#}$rɪŋ#	$\underline{\#}$ræŋ#	Goal: find start of change.
2		0			#$\underline{\text{r}}$ɪŋ#	#$\underline{\text{r}}$æŋ#	
3		3			#r$\underline{\text{ɪ}}$ŋ#	#r$\underline{\text{æ}}$ŋ#	Found.
3	3	3	5	5	#r$\underline{\text{ɪ}}$ŋ#	#r$\underline{\text{æ}}$ŋ#	Goal: find ends of change.
3	4	3	5	5	#r$\underline{\text{ɪ}}$ŋ#	#ræ$\underline{\text{ŋ}}$#	
3	5	3	5	5	#r$\underline{\text{ɪ}}$ŋ#	#ræŋ$\underline{\#}$	
4	3	3	5	5	#rɪ$\underline{\text{ŋ}}$#	#r$\underline{\text{æ}}$ŋ#	
4	4	3	3	3	#rɪ$\underline{\text{ŋ}}$#	#ræ$\underline{\text{ŋ}}$#	Found.

Return ɪ → æ/#r_ŋ#

Application of (91) to ⟨#pʊt#, #pʊt#⟩ (*put/put*).

(93)

i	j	c	d_u	d_v	u	v	Comment
1		0			$\underline{\#}$pʊt#	$\underline{\#}$pʊt#	Goal: find start of change.
2		0			#$\underline{\text{p}}$ʊt#	#$\underline{\text{p}}$ʊt#	
3		0			#p$\underline{\text{ʊ}}$t#	#p$\underline{\text{ʊ}}$t#	
4		0			#pʊ$\underline{\text{t}}$#	#pʊ$\underline{\text{t}}$#	
5		0			#pʊt$\underline{\#}$	#pʊt$\underline{\#}$	No change found.

Return ø → ø/_

Application of (91) to ⟨#θɪŋk#, #θɔt#⟩ (*think/thought*).

(94)

i	j	c	d_u	d_v	u	v	Comment
1		0			#̲θɪŋk#	#̲θɔt#	Goal: find start of change.
2		0			#θ̲ɪŋk#	#θ̲ɔt#	
3		3			#θɪ̲ŋk#	#θɔ̲t#	Found.
3	3	3	6	5	#θɪ̲ŋk#	#θɔ̲t#	Goal: find ends of change.
3	4	3	6	5	#θɪ̲ŋk#	#θɔt̲#	
3	5	3	6	5	#θɪ̲ŋk#	#θɔt#̲	
4	3	3	6	5	#θɪŋ̲k#	#θɔ̲t#	
4	4	3	6	5	#θɪŋ̲k#	#θɔt̲#	
						⋮	
5	3	3	6	5	#θɪŋk̲#	#θɔ̲t#	
						⋮	
6	3	3	6	5	#θɪŋk#̲	#θɔ̲t#	
6	4	3	6	5	#θɪŋk#̲	#θɔt̲#	
6	5	3	5	4	#θɪŋk#̲	#θɔt#̲	Found.
							Return ɪŋk → ɔt/#θ_#

Suffixation Error

The procedure in (95) calculates suffixation errors.

(95)
```
On input $\langle u, v \rangle$:
  1. Let $m$ be the number of segments in $u$ and $n$ the number of segments in $v$. Then
     $u = u_1 \ldots u_m$ and $v = v_n \ldots v_1$.
  2. For $i$ in $1, \ldots, m$:
     (a) For $j$ in $1, \ldots, \min(3, n)$:
         i. If $|u_i \ldots u_m| > |v_n \ldots v_j|$, break out of the inner loop.
         ii. Let $l \leftarrow j - 1 + |u_i \ldots u_m|$.
         iii. If $u_i \ldots u_m = v_l \ldots v_j$, let $S \leftarrow v_{j+1} \ldots v_1$, let $E \leftarrow v_n \ldots v_j$, return the
              error $S/E + \_$.
  3. Return $ø/v + \_$.
```

Application of (95) to ⟨rɪŋ, ræŋ⟩ (*ring/rang*).

(96)

i	j	l	u	v	Comment
1	1	3	$\underline{\text{rɪŋ}}$	$\underline{\text{ræŋ}}$	Goal: match the end of u.
2	1	2	r$\underline{\text{ɪŋ}}$	r$\underline{\text{æŋ}}$	
2	2	3	r$\underline{\text{ɪŋ}}$	$\underline{\text{ræ}}$ŋ	
3	1	1	rɪ$\underline{\text{ŋ}}$	ræ$\underline{\text{ŋ}}$	Found.

Return $ø/[\text{ræŋ}] + \_$

Application of (95) to ⟨lik, likt⟩ (*leak/leaked*).

(97)

i	j	l	u	v	Comment
1	1	3	$\underline{\text{lik}}$	l$\underline{\text{ikt}}$	Goal: match the end of u.
1	2	4	$\underline{\text{lik}}$	$\underline{\text{lik}}$t	Found.

Return $[\text{t}]/[\text{lik}] + \_$

Minimal Generalization Learning

(98) On input $\langle q\_r, q'\_r' \rangle$:

```
1. There are two (possible empty) left environments to be combined, $q = q_m \dots q_1$,
   $q' = q'_n \dots q'_1$. There are two (possibly empty) accompanying right environments
   to be combined, $r = r_1 \dots r_k$, $r' = r'_1 \dots r'_l$.
2. Let $q^g \leftarrow \varepsilon$, the empty string.
3. For $i$ in $1, \dots \min(m,n)$:
    (a) If $q_i = q'_i$, then let $q^g \leftarrow q_i q^g$.
    (b) If $q_i \neq q'_i$, then let $q^g \leftarrow Q_i q^g$, where $Q_i$ is the set of all features contained
        in both $q_i$ and $q'_i$. Stop looping.
4. Let $r^g \leftarrow \varepsilon$, the empty string.
5. For $i$ in $1, \dots \min(k,l)$:
    (a) If $r_i = r'_i$, then let $r^g \leftarrow r^g r_i$.
    (b) If $r_i \neq r'_i$, then let $r^g \leftarrow r^g R_i$, where $R_i$ is the set of all features contained in
        both $r_i$ and $r'_i$. Stop looping.
6. Return $q^g\_r^g$.
```

Application of (98) to ⟨#r_ŋ#, #s_ŋ#⟩.

(99)

i	q	q'	r	r'	q^g	r^g
1	#r̲	#s̲	ŋ̲#	ŋ̲#	[+cons, +cont, +cor, −dor, −lab, −lat, −nas, −syll]	ŋ
2			ŋ#̲	ŋ#̲		ŋ#
					[+cons, +cont, +cor, −dor, −lab, −lat, −nas, −syll]	ŋ#

Application of (98) to ⟨[wɑk] + _, [smæk] + _⟩.

(100)

i	q	q'	q^g
1	wɑk̲	smæk̲	k
2	wɑ̲k	smæ̲k	[−cons, +cont, −high, +low, −nas, +son, +syll, +tns, +vc]k
			[−cons, +cont, −high, +low, −nas, +son, +syll, +tns, +vc]k

Belief Updater

Our Anti-Entropic Constant Reward–Penalty Scheme from section 3.2.2 is given in (101).

$$\begin{array}{rl}
\text{To reward } G_i: & \begin{cases} B_{t+1}(G_i) = B_t(G_i) + \gamma & \text{to a maximum of 1} \\ B_{t+1}(G_j) = B_t(G_j) - \frac{\gamma}{N-1} & \text{for } j \neq i, \text{ to a minimum of 0} \\ N = \text{total number of grammoids} & \end{cases} \\
\text{To penalize } G_i: & \begin{cases} B_{t+1}(G_i) = B_t(G_i) - \gamma & \text{to a minimum of 0} \\ B_{t+1}(G_j) = B_t(G_j) + \frac{\gamma}{N-1} & \text{for } j \neq i, \text{ to a maximum of 1} \end{cases} \\
\text{Otherwise}: & \begin{cases} \text{Stochastically choose a value for } G \text{ (as if we were choosing a grammoid), twice.} \\ -\text{If both values are the same, do nothing.} \\ -\text{If the values differ, stochastically select a third value for } G, \text{ and reward this} \\ \quad \text{grammoid, using } \varepsilon \text{ rather than } \gamma. \end{cases}
\end{array} \tag{101}$$

Several applications of (101), for $\gamma = 0.05$, $\varepsilon = 0.0005$.

(102)

$B(G_1)$	$B(G_2)$	*Then get evidence for*
0	1	*Penalizing G_1*
0	1	*Rewarding G_2*
0	1	*Nothing (then we must choose grammoids; beliefs guarantee that we would choose G_2 twice)*
0	1	*Rewarding G_1*
0.05	0.95	*Rewarding G_1*
0.1	0.9	*Nothing (then suppose we choose different grammoids—there is now an 18% chance of this—and then choose G_2 to be added to—there is a 90% chance of this)*
0.0905	0.9005	*Rewarding G_2*
0.0805	0.9105	*Penalizing G_1*
0.0705	0.9205	*Nothing (then suppose we choose the same grammoid twice—there is now a 13% chance of this)*
0.0705	0.9205	$\vdots$

Morpholexicality Error

The rules in (103) give the error for the morpholexicality of a rule.

(103) **Belief that a rule is morpholexical will be encouraged (morpholexicality deemed good):** ·

If a morpholexical rule skips a form (it would have applied but didn't because the stem was not marked for the rule) and this does not lead to an error; for example, if *lick* is chosen without any rule features, and the morpholexical rule ɪ → ʌ was selected, it will not apply, and this is correct, so we encourage this behaviour.

·If a non-morpholexical rule applies *after* a previous non-morpholexical rule skipped a form (it would have applied but didn't because the stem was marked as an exception to the rule); for example, supposing there are two suffixes, *-d* and -ø, applying in this order, and only the latter is supposed to apply to *ring*, then, if *ring* is a marked exception to the rule, we should also consider the hypothesis that -ø is morpholexical, and we encourage this when we notice that -ø applied because *-d* was skipped over.

·If a non-morpholexical rule applies and there is a rule error in its component (internal change or suffixation); for example, if the ɪ → æ rule applied to *bring* because it was not morpholexical, it would be encouraged to be morpholexical, because it would lead to internal change error.

Belief that a rule is morpholexical will be discouraged (morpholexicality deemed bad): ·

If a morpholexical rule skips a form and this leads to an error with the same change as the rule under consideration; for example, suppose that the suffix *-d* were deemed morpholexical, and it failed to apply to *fry* as a result; then we would discourage it from being morpholexical.

·If a non-morpholexical rule applies and there is no rule error in its component; for example, if the suffix *-d* applies to *fry* and yields *fried*, the rule's morpholexicality will be (further) discouraged.

Use of (103) on a derivation for *put* ([pʊt]), with an exception feature for rule 2. Starred rules are morpholexical.

(104)

	Input to Suffixation	[pʊt] ([−2])	
(1)	[ɪd]/[+ant, +cor, −son, −cont] + _ (*)	*Skips*	*Morph'ty good*
(2)	[t]/[−voice, −son] + _	*Skips due to exclusion feature*	*No evidence*
(3)	ø/[+cor] + _	*Applies after previous rule skipped*	*Morph'ty good*
(4)	[d]/ + _	—	*No evidence*
	Suffixation Error	*None*	

Use of (103) on a derivation for *wait* ([wet]), with no stem markings selected. Starred rules are morpholexical.

(105)

	Input to Suffixation	[wet]	
(1)	[ɪd]/[+ant, +cor, −son, −cont] + _ (*)	*Skips*	*Morph'ty bad*
(2)	[t]/[−voice, −son] + _	*Applies*	*Morph'ty good*
(3)	ø/[+cor] + _	—	*No evidence*
(4)	[d]/ + _	—	*No evidence*
	Suffixation Error	[ɪd]/[wet] + _	

Use of (103) on a derivation for *put* ([pʊt]), with no stem markings. Starred rules are morpholexical.

(106)

	Input to Suffixation	[pʊt]	
(1)	[ɪd]/[+ant, +cor, −son, −cont] + _ (*)	*Skips*	*Morph'ty bad*
(2)	[t]/[−voice, −son] + _	*Applies*	*Morph'ty good*
(3)	ø/[+cor] + _	—	*No evidence*
(4)	[d]/ + _	—	*No evidence*
	Suffixation Error	ø/[pʊt] + _	

Use of (103) on a derivation for *beg* ([bɛg]), with no stem markings. Starred rules are morpholexical.

(107)

	Input to Suffixation	[bɛg]	
(1)	[ɪd]/[+ant, +cor, −son, −cont] + _ (*)	*Skips*	*Morph'ty good*
(2)	[t]/[−voice, −son] + _	*Does not apply*	*No evidence*
(3)	ø/[+cor] + _	*Does not apply*	*No evidence*
(4)	[d]/ + _	*Applies*	*Morph'ty bad*
	Suffixation Error	*None*	

Stem Marking Error

The rules in (108) give the error for inclusion or exception features on a stem for some rule.

(108) **Inclusion good:** Morpholexicality was deemed *bad*.

Inclusion bad: Morpholexicality was deemed *good*.

Exception good: Morpholexicality was deemed *good*.

Exception bad: Morpholexicality was deemed *bad*.

Use of (108) on a derivation for *put* ([pʊt]), with no stem markings. Starred rules are morpholexical.

(109)

	Input to Suffixation		[pʊt]
(1)	ø/[+cor] + _ (*)	*Skips*	*Morph'ty bad, evidence for* [+1]
(2)	[ɪd]/[+ant, +cor, −son, −cont] + _	*Applies*	*Morph'ty good, evidence for* [−2]
(3)	[t]/[−voice] + _	—	*No evidence*
(4)	[d]/ + _	—	*No evidence*
	Suffixation Error		ø/[pʊt] + _

Use of (108) on a derivation for *put* ([pʊt]), with an exception for rule 2. Starred rules are morpholexical.

(110)

	Input to Suffixation		[pʊt] ([−2])
(1)	ø/[+cor] + _ (*)	*Skips*	*Morph'ty bad, evidence for* [+1]
(2)	[ɪd]/[+ant, +cor, −son, −cont] + _	*Skips due to exclusion feature*	*No evidence*
(3)	[t]/[−voice] + _	*Applies*	*Morph'ty good, evidence for* [−3]
(4)	[d]/ + _	—	*No evidence*
	Suffixation Error		ø/[pʊt] + _

Use of (108) on a derivation for *put* ([pʊt]), with an inclusion for rule 1. Starred rules are morpholexical.

(111)

	Input to Suffixation	[pʊt] ([+1])	
(1)	ø/[+cor] + _ (*)	*Applies*	*No evidence*
(2)	[ɪd]/[+ant, +cor, −son, −cont] + _	—	*No evidence*
(3)	[t]/[−voice] + _	—	*No evidence*
(4)	[d]/ + _	—	*No evidence*
	Suffixation Error	*None*	

Rule Existence Error

The rules in (112) give the error for the existence lesson for some rule.

(112) **Existence good:**

The rule applied, and there was no error in its component.

Existence bad:

The rule applied, and there was an error in its component; the rule was skipped over (because it was morpholexical or the stem was marked with an exception feature) but there was no error, and a rule with the same change applied.

Use of (112) on a derivation for *receive* ([rəsiv]), with an exception for rule 5. Starred rules are morpholexical.

(113)

	Input to Suffixation	[rəsiv] ([−5])	
(1)	[d]/[+cont, −dor, −lat, +son, +voice] + _ (*)	*Skips*	*Existence bad*
(2)	[ɪd]/[[+cont, −dor, −lab, −lat, −nas, +son, −str, +voice]t] + _	*Does not apply*	*No evidence*
(3)	[t]/[−lab, −lat, −nas, −son, −voice] + _	*Does not apply*	*No evidence*
(4)	[ɪd]/[+ant, +cor, −son, −cont] + _	*Does not apply*	*No evidence*
(5)	[t]/[−nas] + _	*Skips*	*No evidence*
(6)	[d]/ + _	*Applies*	*Existence good*
	Suffixation Error	*None*	

Use of (112) on a derivation for *receive* ([rəsiv]), with no stem markings. Starred rules are morpholexical.

(114)

	Input to Suffixation	[rəsiv]	
(1)	[d]/[+cont, −dor, −lat, +son, +voice] + _ (*)	*Skips*	*No evidence*
(2)	[ɪd]/[[+cont, −dor, −lab, −lat, −nas, +son, −str, +voice]t] + _	*Does not apply*	*No evidence*
(3)	[t]/[−lab, −lat, −nas, −son, −voice] + _	*Does not apply*	*No evidence*
(4)	[ɪd]/[+ant, +cor, −son, −cont] + _	*Does not apply*	*No evidence*
(5)	[t]/[−nas] + _	*Applies*	*Existence bad*
(6)	[d]/ + _	—	*No evidence*
	Suffixation Error	[d]/[rəsiv] + _	

Appendix E

Feature Chart

	ant	back	cons	cont	cor	dor	high	lab	lat	low	nas	round	son	str	syll	tns	vc
ɑ		+	−	+			−			+	−	−	+		+	+	+
ʌ		+	−	+			−			−	−	−	+		+	+	+
æ		−	−	+			−			+	−	−	+		+	+	+
ɛ		−	−	+			−			−	−	−	+		+	−	+
e		−	−	+			−			−	−	−	+		+	+	+
ɪ		−	−	+			+			−	−	−	+		+	−	+
i		−	−	+			+			−	−	−	+		+	+	+
u		+	−	+			+			−	−	+	+		+	+	+
ʊ		+	−	+			+			−	−	+	+		+	−	+
o		+	−	+			−			−	−	+	+		+	+	+
ɔ		+	−	+			−			+	−	+	+		+	+	+
ŋ	−		+	+	−	+		−	−		+		+	−	−		+
n	+		+	+	+	−		−	−		+		+	−	−		+
m	+		+	+	−	−		+	−		+		+	−	−		+
j	+		−	+	+	−		−	−		−		+	−	−		+
l	+		+	+	+	−		−	+		−		+	−	−		+
r	−		+	+	+	−		−	−		−		+	−	−		+
w	−		−	+	−	+		+	−		−		+	−	−		+
k	−		+	−	−	+		−	−		−		−	−	−		−
g	−		+	−	−	+		−	−		−		−	−	−		+

	ant	back	cons	cont	cor	dor	high	lab	lat	low	nas	round	son	str	syll	tns	vc
h	−		+	+	−	+		−	−		−		−	−	−		−
č	−		+	−	+	−		−	−		−		−	+	−		−
đ	−		+	−	+	−		−	−		−		−	+	−		+
ʃ	−		+	+	+	−		−	−		−		−	+	−		−
ʒ	−		+	+	+	−		−	−		−		−	+	−		+
t	+		+	−	+	−		−	−		−		−	−	−		−
d	+		+	−	+	−		−	−		−		−	−	−		+
s	+		+	+	+	−		−	−		−		−	+	−		−
z	+		+	+	+	−		−	−		−		−	+	−		+
θ	+		+	+	+	−		−	−		−		−	−	−		−
ð	+		+	+	+	−		−	−		−		−	−	−		+
p	+		+	−	−	−		+	−		−		−	−	−		−
b	+		+	−	−	−		+	−		−		−	−	−		+
f	+		+	+	−	−		+	−		−		−	−	−		−
v	+		+	+	−	−		+	−		−		−	−	−		+

Bibliography

ALBRIGHT, ADAM. 2002. Islands of reliability for regular morphology: Evidence from Italian. *Language* 78.684–709.

——, in prep. Natural classes are not enough: Biased generalization in novel onset clusters. Unpublished manuscript, Massachussets Institute of Technology (http://web.mit.edu/albright/www/papers/Albright-BiasedGeneralization.pdf).

——, and BRUCE HAYES, 1999. An automated learner for phonology and morphology. Unpublished manuscript, UCLA (http://www.linguistics.ucla.edu/people/hayes/learning/learner.pdf).

——, and BRUCE HAYES. 2002. Modeling English past intuitions with Minimal Generalization. In *Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*. Association for Computational Linguistics.

——, and BRUCE HAYES. 2003. Rules vs analogy in English past tenses: A computational/experimental study. *Cognition* 90.119–161.

——, and BRUCE HAYES. 2006. Modeling productivity with the Gradual Learning Algorithm: The problem of accidentally exceptionless generalizations. In *Gradience in Grammar: Generative Perspectives*, ed. by Gilbert Fanselow, Caroline Fery, Matthias Schlesewsky, and Ralf Vogel. Oxford: Oxford University Press.

ALISHAHI, AFRA, 2008. *A Probabilistic Model of Early Argument Structure Acquisition*. University of Toronto dissertation.

BARTKE, SUSANNE, FRANK RÖSLER, JUDITH STREB, and RICHARD WIESE. 2005. An ERP-study of German ‘irregular’ morphology. *Journal of Neurolinguistics* 18.29—-55.

BERKO, JEAN. 1958. The child's learning of English morphology. *Word* 14.150–177.

BIRD, HELEN, MATTHEW LAMBON RALPH, MARK SEIDENBERG, JAMES MCCLELLAND, and KARALYN PATTERSON. 2003. Deficits in phonology and past-tense morphology: What's the connection? *Journal of Memory and Language* 48.502–526.

BLOOM, LOIS. 1973. *One Word At A Time: The Use of Single Word Utterances Before Syntax*. Mouton.

——, LUCY HOOD, and PATSY LIGHTBOWN. 1974. Imitation in language development: If, when and why. *Cognitive Psychology* 6.380–420.

BOHANNON, JOHN, and ANN MARQUIS. 1977. Children's control of adult speech. *Child Development* 48.1002–1008.

BROWN, ROGER. 1973. *A First Language: The Early Stages*. Harvard University Press.

BYBEE, JOAN. 2006. From usage to grammar: The mind's response to repetition. *Language* 82.711–733.

——, and CAROL LYNN MODER. 1983. Morphological classes as natural categories. *Language* 59.251–270.

——, and DAN SLOBIN. 1982. Rules and schemes in the development and use of the English past tense. *Language* 58.265–289.

CHAN, ERWIN, 2008. *Structures and Distributions in Morphology Learning*. University of Pennsylvania dissertation.

CHOMSKY, NOAM. 1957. *Syntactic Structures*. The Hague: Mouton.

——. 1965. *Aspects of the Theory of Syntax*. Cambridge: MIT Press.

——, and MORRIS HALLE. 1968. *The Sound Pattern of English*. New York: Harper and Row.

——, and HOWARD LASNIK. 1977. Filters and control. *Linguistic Inquiry* 8.425–504.

CLARK, ROBIN. 1992. The selection of syntactic knowledge. *Language Acquisition* 2.83–149.

DAUGHERTY, KIM, and MARK SEIDENBERG. 1992. Rules or connections? The past tense revisited. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 259–264, Hillsdale, NJ. Erlbaum.

EMBICK, DAVID, and ALEC MARANTZ. 2005. Cognitive neuroscience and the English past tense: Comments on the paper by Ullman et al. *Brain and Language* 93.243–247.

FODOR, JANET DEAN. 1998. Unambiguous triggers. *Linguistic Inquiry* 29.1–36.

GILDEA, DANIEL, and DANIEL JURAFSKY. 1995. Automatic induction of finite state transducers for simple phonological rules. In *Proceedings of ACL 95*, 9–15.

GOLDSMITH, JOHN. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics* 27.153–198.

HALL, WILLIAM, WILLIAM NAGY, and ROBERT LINN. 1984. *Spoken Words: Effects of Situation and Social Group on Oral Word Usage and Frequency*. Lawrence Erlbaum.

HALLE, MORRIS, and ALEC MARANTZ. 1993. Distributed Morphology and the pieces of inflection. In *The View from Building 20*, ed. by Ken Hale and Samuel J. Keyser. Cambridge: MIT Press.

——, and K. P. MOHANAN. 1985. Segmental phonology of Modern English. *Linguistic Inquiry* 16.57–116.

HIGGINSON, ROY, 1985. *Fixing-Assimilation in Language Acquisition*. University of Washington dissertation.

JOANISSE, MARC, and MARK SEIDENBERG. 1999. Impairments in verb morphology after brain injury: A connectionist model. *Proceedings of the National Academy of Sciences* 96.7592—-7597.

KONDRAK, GRZEGORZ, 2002. *Algorithms for Language Reconstruction*. University of Toronto dissertation.

KUCZAJ, STAN. 1977. The acquisition of regular and irregular past tense forms. *Journal of Verbal Learning and Verbal Behaviour* 16.589–600.

KUČERA, HENRY, and W. NELSON FRANCIS. 1967. *Computational Analysis of Present-Day English*. Providence, RI: Brown University Press.

LIGHTFOOT, DAVID. 1982. *The Language Lottery*. Cambridge: MIT Press.

——. 1989. The child's trigger experience: Degree-0 learnability (target article). *Behavioral and Brain Sciences* 12.321–334.

——. 1991. *How to Set Parameters: Arguments From Language Change*. Cambridge: MIT Press.

LING, CHARLES. 1994. Learning the past tense of English verbs: The Symbolic Pattern Associator vs connectionist models. *Journal of Artificial Intelligence Research* 1.209–229.

——, and MARIN MARINOV. 1993. Answering the connectionist challenge: A symbolic model of learning the past tenses of English verbs. *Cognition* 49.235–290.

MACWHINNEY, BRIAN. 2000. *The CHILDES Project: Tools for Analyzing Talk.*. Mahwah, NJ: Lawrence Erlbaum Associates, Third edition.

——, and JARED LEINBACH. 1991. Implementations are not conceptualisations: Revising the verb learning model. *Cognition* 29.121–157.

MARCUS, GARY, STEVEN PINKER, MICHAEL ULLMAN, MICHELLE HOLLANDER, T. JOHN ROSEN, and FEI XU. 1992. *Overregularization in Language Acquisition.* Chicago: University of Chicago Press.

MAYOL, LAIA. 2003. Acquisition of irregular patterns in Spanish verbal morphology. In *Proceedings of the Twelfth ESSLLI Student Session*, ed. by Ville Nurmi and Dmitry Sustretov.

MOLNAR, RAYMOND, 2001. “Generalize and Sift” as a model of inflection acquisition. Master’s thesis, Massachussets Institute of Technology.

NARENDRA, K., and M. THATACHAR. 1989. *Learning Automata*. Englewood Cliffs, NJ: Prentice-Hall.

NOYER, ROLF. 1997. *Features, Positions, and Affixes in Autonomous Morphological Structure*. New York: Garland. Published version of 1992 dissertation.

PATTERSON, KARALYN, MATTHEW LAMBON RALPH, JOHN HODGES, and JAMES MCCLELLAND. 2001. Deficits in irregular past-tense verb morphology associated with degraded semantic knowledge. *Neuropsychologia* 39.709–724.

PINKER, STEVEN. 1991. Rules of language. *Science* 253.530–535.

——. 1999. *Words and Rules*. New York: Harper and Row.

——, and ALAN PRINCE. 1988. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition* 28.73–193.

——, and ALAN PRINCE. 1994. Regular and irregular morphology and the psychological status of rules of grammar. In *The Reality of Linguistic Rules*, ed. by Susan Lima, Roberta Corrigan, and Gregory Iverson, 321–351. Philadephia: John Benjamins.

PLUNKETT, KIM, and PATRICK JUOLA. 1999. A connectionist model of English past tense and plural morphology. *Cognitive Science* 23.463–490.

——, and VICTORIA MARCHMAN. 1993. From rote learning to system building: Acquiring verb morphology in children and connectionist nets. *Cognition* 48.21–69.

PRASADA, SANDEEP, and STEVEN PINKER. 1993. Generalisation of regular and irregular morphological patterns. *Language and Cognitive Processes* 8.1–56.

RUMELHART, DAVID, and JAMES MCCLELLAND. 1986. On learning the past tenses of English verbs. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Volume 2: Psychological and Biological Models*, ed. by James McClelland and David Rumelhart, volume 2. Cambridge: MIT Press.

SACHS, JACQUELINE. 1983. Talking about there and then: The emergence of displaced reference in parent-child discourse. In *Children's Language: Volume 4*, ed. by Keith Nelson. Lawrence Erlbaum.

SCHÜTZE, CARSON. 2005. Thinking about what we are asking speakers to do. In *Linguistic evidence: Empirical, theoretical, and computational perspectives*, ed. by Stephan Kepser and Marga Reis, 457–485. Berlin: Mouton de Gruyter.

THORNTON, ROSALIND, and GRACIELA TESAN, 2007. Parameter setting and statistical learning. Unpublished manuscript, Macquarie University.

ULLMAN, MICHAEL, SUZANNE CORKIN, MARIE COPPOLA, GREGORY HICKOK, JOHN GROWDON, WALTER KOROSHETZ, and STEVEN PINKER. 1997. A neural dissociation within language. *Journal of Cognitive Neuroscience* 9.266–276.

——, ROUMYANA PANCHEVA, TRACY LOVE, EILING YEE, DAVID SWINNEY, and GREGORY HICKOK. 2005. Neural correlates of lexicon and grammar: Evidence from the production, reading, and judgment of inflection in aphasia. *Brain and Language* 93.185–238.

WAGNER, ROBERT, and MICHAEL FISHER. 1974. The string-to-string correction problem. *Journal of the Association for Computing Machinery* 21.168–173.

WINITZ, HARRIS, ROBERT SANDERS, and JOAN KORT. 1981. Comprehension and production of the /-ez/ plural allomorph. *Journal of Psycholinguistic Research* 10.259–271.

XU, FEI, and STEVEN PINKER. 1995. Weird past tense forms. *Journal of Child Language* 22.531–56.

YANG, CHARLES. 2002. *Knowledge and Learning in Natural Language*. Oxford: Oxford University Press.

——. 2005. On productivity. *Linguistic Variation Yearbook* 5.265–302.

YIP, KENNETH, and GERALD JAY SUSSMAN. 1997. Sparse representations for fast, one-shot learning. In *Proceedings of National Conference on Artificial Intelligence*.